Reconstructing Distribution from Option Prices

Jacky Sung

Supervised by Dr. Rua Murray and Dr. Carl Scarrott

A thesis submitted in fulfilment of the requirements of the Degree for Master of Science in Mathematics at the University of Canterbury

April 2013.

Abstract

Option pricing has been a popular topic in the financial industry. If there were an effective way to price options correctly, it could help to identify potential profits and risks in the options market. The risk-neutral density (RND), if it exists, leads to calculation of fair prices of options by taking the expected value of the payoff function under the RND, which can be reconstructed non-parametrically from the market data alone. Jaynes (1963) argues that, out of the infinitely many density functions, there is a unique and most preferred way to choose the density: which is via the maximum entropy principle, and hence, the density obtained is called maximum entropy density (MED). A classical approach of finding the MED is by maximising the Lagrangian function with Lagrange multipliers; however, due to potential numerical difficulties, this is reformulated under the duality result by Borwein et al. (2003).

This thesis carries out a simulation study to explore the properties of the MED estimators proposed in the literature. With the framework given, some data were simulated with a log-normal distribution and found that the MED constructed converges to the original distribution when the data is convex and noiseless. However, it is inevitable for the market data to be noisy, simulated noise is added and explores effective methods that would not only filter out the noise but also guarantee the existence of MED were explored.

Many possible strategies that deal with noisy and non-convex data including the Tikhonov regularisation, polyhedral set projection, convex hull methods, and the cubic spline smoothing methods have been attempted. As a result of 1000 replications of simulated experiments, the cubic spline smoothing method outperforms the other methods yielding the lowest mean integrated squared error. Some of the reconstructed densities give relatively accurate results. This method was then applied to real VIX indices data, the results obtained, however, depended on the choice of the mixing parameter p, which could be subjective at times.

Acknowledgements

Firstly I would like to give my utmost thank to my senior supervisor Dr. Rua Murray for offering me this opportunity to continue with my mathematics expedition. At times when I was lost and buried in frustration, it is always Rua who have brought me the light to the exit. Next I would like to thank my co-supervisor Dr. Carl Scarrott for being an extremely supportive mentor and friend who also supervised my honours project in 2010. Carl has been providing constant help with statistical advices and late night proofreading this thesis. Another big thank must go to Dr. Marco Reale for his encouragement and mental support since I started this project. It is him who motivates me and let me feel that I am a really good person.

I would like to thank my parents for loving, supporting, and believing me with all their hearts. It is them who let me believe there is unconditional love in this world. Also I would like to thank my lovely little sister for taking care of me at times.

I would also like to thank all my friends in the Maths and Stats department, specifically Joe Zhu and Paul Brouwers for leading me to the wonderful SVN so I can work anywhere with backups and version controls.

Special thank must go to my special friend Jackie Lee, it is him who have constantly given me the motivation to strive forward in these two years, providing me with many smart laughter while I was down, and giving me full support when I was doubting myself. Another big thank must go to Sharon Yu for proofreading my thesis and letting me know it is not difficult to write in English, but not easy to make it right. Finally, I would also like to thank all my friends who have been supporting and understanding me all the way through, I know I have been hiding from you guys while I was busy typing this thesis up. I am very certain without any of those people, there is no way this thesis can be finished on time.

Contents

Chapter 1

Introduction

Having uncertainties in future events often causes fear. This is often the case if the outcome turned out to be opposite to people's expectation, which could sometimes lead to huge losses in their life. In the stock trading world, stock price movements are the key factor for all decisions. For example, someone would want to buy and hold a stock when its price is rising, and sell it in the market when the price gets even higher. This is what many analysts do everyday - projecting the future stock prices in the market - but it is the investors who will need to bear the risk and uncertainty when the projections go wrong. One of the ways to reduce the level of uncertainty in stock trading is through the invention of 'option' contracts, which gives an option owner a right to trade a stock at a fixed price in a future date. An option holder can decide whether he/she wants to exercise his/her right to trade with this agreed price on the option contract, which is also called the "exercise" price or the "strike" price (these two terms will be used interchangeably throughout this thesis). Because of this, option owners can act according to the way stock prices move in the future: one would buy a stock with an option if the actual stock price is more expensive in the future than the agreed exercise price specified in the option; or on the contrary would short a stock with the option when the future price has dropped. This way, uncertainty or potential losses that one can have who based on the future prices can be effectively controlled at a low level.

Call options are a type of financial derivative or contract which gives the buyer the right, but not the obligation, to purchase a stock at a 'maturity' time T with strike price k. The stock on whose price the contract is written is called the "underlying". Put options are defined similarly but gives the right to sell the underlying stock. Options can be traded at option exchanges or the over-the-counter market at a premium which is the price that the buyer must pay (see [19]). There are two types of "vanilla" options: the American and the European option. The American option can be exercised at any time before the maturity date whereas the European option must be exercised at maturity. Because of this, it is expected that the American option will be charged at a flexible cost, but somehow higher than the European options due to the extra right it possesses. In this thesis, only the European call options will be studied.

Option pricing theory is based on a principle called "arbitrage-free principle". Arbitrage-free principle is a fundamental concept: one should not be able to make riskless profits. More precisely, there should be no guaranteed way to make a non-zero profit with initial zero cash flows. The opportunity that would guarantee the investors to create such non-zero profit is known as the Type A arbitrage, or the absolute arbitrage. In addition, the Type B arbitrage, also known as the statistical arbitrage, is the opportunity to make a profit with negative initial costs (see [17] and [4]). Arbitrage opportunities, however, do exist in the real market as a result of the market's perception that there exists a deviation from the historical average of the securities in a statistically significant way [12]. One of the ways is to construct a strategy to trade in different underlying securities (for instance, a strategy that involves buying a certain amount of calls and selling some puts) and earn from the price differences without bearing any risks! Arbitrage opportunities are rare and will quickly disappear in about one second once they are exploited [3]. Therefore, if we assume we have an arbitrage-free environment, we would be able to price every underlying and derivative (such as options) in a fair and correct way such that no one would be able to gain arbitrage from it. So we will need to comply with this arbitrage-free principle in order to price options as "fair prices".

The Black-Scholes option pricing model appeared in 1973 and was proposed by Black, Scholes and Merton ([5] and [23]). They gave a way of pricing options by forming a partial differential equation (now known as the Black-Scholes equation, see details in [5]).

The following are the assumptions that they used for their equation:

1. Has to comply with the arbitrage-free principle
2. The price of the underlying follows a geometric Brownian motion, so the stocks will be distributed with a log-normal distribution with constant drift μ and volatility σ over time
3. Needs options to be traded at risk free interest rates
4. Needs to be able to buy or sell any amount (or fractional) of stocks
5. Has zero transaction costs
6. As no dividend paid by the stocks

A direct result of the Black-Scholes equation is the following Black-Scholes formula, which gives the fair price of an European call option C as an expectation of a log-normal density (which is the risk-neutral density of the geometric Brownian motion):

Black-Scholes Formula:

$$C = N(d_1)S - N(d_2)K\exp\left(-r(T-t)\right) \tag{1.1}$$
$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}$$
$$d_2 = d_1 - \sigma\sqrt{T-t}$$

where S is the current spot price of the underlying stock, $N(\cdot)$ the normal cumulative distribution function, K the strike price of the option, T the maturity date, t the current time (then $T-t$ is just the time to maturity),

This equation was a big breakthrough at that time, earning Scholes and Merton Nobel prizes in Economics in 1997. This approach provides a benchmark price for options in the real market. It is convenient for the option buyers to make a decision by observing how much deviation the market price is from the Black-Scholes price. However, there are some drawbacks to the model. One of the requirements stating that the implied prices must equal to its actual value to prevent arbitrage opportunities was violated (see [22]). Longstaff in [22] performed an empirical test and found that the martingale restriction imposed by the Black-Scholes model was strongly rejected by the S&P market data. Figlewski [14] also discussed about how this arbitrage-based model cannot be used to explain market price in times of financial crisis. He also discussed the assumptions of this model failing to hold in the real world: volatility is never known. Moreover, Rubinstein [29] found that implied volatility (a volatility measure induced by the observed prices) is not even constant but exhibiting a curve shape over time (hence it is sometimes referred to as "volatility smile", see Hull [19] p381 - 392).

From above, we see that Black-Scholes prices may not be the best to represent fair value of the market data; yet it is still widely used by many people in spite of the mispricing. Many others have tried to come up with alternative option pricing models to compare too. Since it was mentioned that the volatility was not constant as assumed by Black and Scholes, many people have then calibrated the Black-Scholes model with various volatility structures such as conditional volatility models, realised volatility models, and stochastic volatility models, etc. Currently researches are focussing on stochastic volatility models in particular, Heston model [18] and SABR model [16] have been discussed most frequently.

An alternative way to price the options is to first to find the **risk-neutral distributions** (RND) of the option prices. From this distribution, we were able to infer the fair prices via the expected value of the future payoff generated by the options (see [28] and [11]). That is, the distribution of the option prices is looked at instead of the actual stock price movement - this is the approach taken in this thesis. By having the market option prices as the dataset, then the distribution will be reconstructed from it.

Herzel in [17] showed that arbitrage-free principle, where our risk-neutral property was based on,

complies if and only if the function of the observed option prices against the strike prices is strictly convex (a notion will be introduced in Chapter 2), and strictly decreasing with corresponding increasing strike prices (see Theorem 4.1 in [17]). Hence we will need the option prices to be convex in order for RND to be valid. A set of convex option prices will be continuous and twice-differentiable. Breeden and Litzenberger [9] have shown that the RND is proportional to the second derivative of the option price function with respect to the strike prices (so RND is non-negative as we know that the second derivatives of any convex function will be also non-negative). Because of this promising result, recovering the RND from observed prices becomes possible. Many non-parametric methods were then used to model the RND such as the positive convolution approximation [6], cubic spline method [24], implied Binomial models [29] and [20]. The least prejudiced non-parametric model (with respect to the missing data) is via the **maximum entropy principle** (see Jaynes [21]).

The maximum entropy principle is a way to estimate the distribution of a random variable relying solely on the information given. The density and solution with the largest entropy are called the **maximum entropy density** (MED) and **maximum entropy solution** (MES), respectively. Jaynes [21] argues that the distribution which has the largest entropy is preferred since it will be maximally noncommittal with respect to missing or unknown information. For instance, if we have no information or data at all, the MES on a finite interval will be a uniform distribution (assuming not one point has a higher chance than another); but if we have only the mean then the MES obtained on $(0, \infty)$ will be an exponential distribution; and if we have both the mean and the variance the MES on the interval $(-\infty, \infty)$ will be a normal distribution (see [10]). From here we can see the MES induced is sensible and natural. So if we only have limited information about a random variable then the best we can say about the distribution is the realisation of those random variables, and in our case the option prices. Therefore, the MED will be the best distribution we can construct by relying on just the prices. Buchen and Kelly [10] were one of the first to apply the maximum entropy principle to option prices.

Generally, a maximum entropy problem will with

Maximum Entropy Problem

$$\text{Find} \quad \sup_p \; -\int_I \phi(p(x))\, dx$$

subject to

$$1 = \int_I p(x)\, dx$$

$$\mu_i = E(f_i) = \int_I f_i(x) p(x)\, dx \qquad \text{for } i = 1, \ldots, n.$$

where ϕ is the entropy functional, I the domain, n the number of known moments.

So in our application of MEP for option pricing, let us first define the payoff function c_i for each

strike price k_i:

$$c_i(x) = (x - k_i)_+ = \begin{cases} x - k_i & \text{if } x \geq k_i, x \in I \\ 0 & \text{otherwise.} \end{cases}$$

With this and a more general entropy function ϕ and μ_i is just the option prices d_i. The problem is now:

Maximum Entropy Problem for Option Pricing

$$\text{Find} \quad \sup_p \; -\int_0^\infty \phi(p(x))\, dx \tag{1.2}$$

subject to

$$1 = \int_0^\infty p(x)\, dx \tag{1.3}$$

$$d_i = \int_0^\infty c_i(x) p(x)\, dx \qquad \text{for } i = 1, \ldots, m. \tag{1.4}$$

where $x \in I$; I is the interval of the feasible prices; $p(\cdot)$ the risk-neutral density.

This optimisation problem has two constraints (1.3) and (1.4): the first constraint ensures that the RND is a probability function that integrates to one over all support of x and the second constraint is an indexed family of risk-neutral properties, having one for each data point. That is, there will be no risk if the option is priced by $d_i = E_p[c_i]$, the expected value of the payoff function under the density $p(x)$ (see [2]).

The most common way to solve this maximising problem with the given constraints is by using the Lagrangian, which has the following form:

Lagrangian

$$\text{Find} \quad \inf_\lambda \sup_p L(\lambda, p)$$

where

$$L(\lambda, p) := -\int_0^\infty p(x) \ln p(x)\, dx + \lambda_0 \left(1 - \int_0^\infty p(x)\, dx\right) + \sum_{i=1}^m \lambda_i \left(d_i - \int_0^\infty c_i p(x)\, dx\right). \tag{1.5}$$

Avellaneda [1] then reduced the Lagragian to the form:

$$\inf_\lambda \left(\ln Z(\boldsymbol{\lambda}) - \sum_{i=1}^m \lambda_i d_i\right), \tag{1.6}$$

where

$$Z(\lambda) = \int_0^\infty e^{\sum_{i=1}^m \lambda_i c_i(x)}\, dx,$$

By setting the first order partial derivatives to 0 in (1.6), we obtain λ^* as the solution which maximises the Lagrangian problem. The MES, $p^*(x)$ obtained has a form of:

$$p(x) = \frac{\exp\left[\sum_{i=1}^m \lambda_i^* c_i(x)\right]}{\int_0^\infty \exp\left[\sum_{i=1}^m \lambda_i^* c_i(x)\right] dx}. \tag{1.7}$$

where $c_i(x)$ is the payoff function of x at strike price k_i for $x \in I$, λ^* are the Lagrange multiplier solution that maximises the entropy over m strike prices.

However, there are some flaws to this Lagrangian approach as Borwein [7] pointed out: the value function will not be differentiable and continuous on the complement of $L^1(I)$. In order to legitimise the calculations in (1.7), Borwein attempted to remedy this problem via convex duality. They successfully turned the problem into a finite dimensional problem and also discovered that the MES exists if and only if the observed data lie within an open polyhedral set formed by the constraints (1.3) and (1.4). This is the **constraint qualification** (CQ). The MES they obtained with the dual problem turns out to be the same as (1.7).

Orozco-Rodriguez and Santosa [26] then continued the work that Buchen, Kelly and Borwein et. al have left off. They first simulated a data set from the log-normal distribution, then performed various experiments looking at the conditioning of the Hessian matrix of the dual value function and the solution where noise is present in the data. They suggested two strategies to deal with the noise: first is to project the noisy data onto the relative interior of the open polyhedral set formed by the constraints. They achieved this by fixing a small ϵ and turned the conditions into a bounded ones (this will be stated in chapter 4) and the other strategy is to regularise the dual problem with the Tikhonov regularisation. The regularisation turned out to be very successful and the reconstructed density is also smooth when the regularisation parameter α is picked adequately.

Although it is convenient to use maximum entropy for density estimation, it is easy to have an ill-conditioned problem. This is because when more and more price constraints are added in it which might cause the optimisation to run into difficulties as the Jacobian matrix of the value function would become more ill-conditioned (see [10]). This problem was then addressed by Orozco-Rodriguez and Santosa in [26]. However, Bose and Murray [8] tackled and reformulated this problem with a different set of basis function. Instead of looking at the infinite support the payoff function $c_i(x)$ has (in 1.4), they chose a hat function $h(x)$, such that it has finite support over the feasible intervals. The hat functions also provide a way to measure convexity violation through a moment they introduced called "$\boldsymbol{\eta}$" defined by

$$\boldsymbol{\eta} = \int_0^K h(x)p(x)\,dx.$$

From this formulation, as $h(x)$ and $p(x)$ are always non-negative, $\boldsymbol{\eta}$ needs to be non-negative too. When η_i is zero in the interval (k_{i-1}, k_{i+1}), implying the data point in this interval is non-strictly convex. Hence, this suggest the interval (k_{i-1}, k_{i+1}) to be removed from the domain to make sure the whole domain is feasible, which prevents computation difficulties. For the union of those (k_{i-1}, k_{i+1}) intervals that correspond to $\eta_i = 0$ is called the "impossible prices interval".

It is already evident that option pricing is important so that risks and uncertainties could be reduced. However, the Black-Scholes model does not price the options correctly, which leads to the incident of maximum entropy model. The model is totally data driven, hence will be sensitive to any noise in the data. This thesis extends Bose and Murray's work to explore an effective strategy that deals with noise and convexity violation in the data set to make sure the reconstruction process is valid.

Many strategies have been explored and studied, including

- Tikhonov regularisation: A method in [26] which adds a penalty function of λ to the value function being optimised, which can effectively remedy the diagonals of the Hessian matrix when it is ill-conditioned.
- Polyhedral set projection: A method in [26] which applies a projection of the option price data onto the polyhedral set formed by the CQ. This method allows the projected data be convex and guaranteeing a maximum entropy solution to exist, but could easily have an ill-conditioned Hessian matrix when the data lie closely to the boundary of the polyhedral set.
- Convex hull method: A method I applied in this thesis, finding a convex subset of the existing data set. This method will guarantee the data set be convex and satisfying CQ, however, does not account for the noise reduction.
- Data linearising: A new method which sets the negative η_i's to zero in a way that adjusts the neighbouring η_i's and also retaining the condition of the sum of $\boldsymbol{\eta}$ being one.
- Cubic spline smoothing method: A method I applied in this thesis, inspired by [15] and [24], which preprocesses the data by fitting a cubic spline smoothing curve through it such that it could smooth out the noise level and guarantees the data to be convex.

Moreover, the conditioning of the Hessian matrix will be studied in order to understand why the optimisation fails as a result of non-positive η_i's. It is also shown that Tikhonov regularisation is

useful to remedy ill-conditioned problems.

This thesis is organised in the following manner. Chapter 2 contains all the definitions and minimal basic theories in order to understand this thesis. Chapter 3 uses the data sets Orozo-Rodriguez & Santosa used in [26] and applies the maximum entropy principle to reconstruct the densities from noiseless data, then the simulation of noise will be added to this data. Chapter 4 tests the strategies previous mentioned on 1000 simulated randomly generated noisy data set with the minimal background theories of each strategy included. Chapter 5 applies the cubic spline smoothing method onto three pieces of VIX data in the real market. Chapter 6 provides a thorough analysis on how conditioning is affected by small η_i's. Finally the conclusion is summarised in Chapter 7 and followed by the Appendix and the references. Note that all of the programs in MATLAB in this project were all written by myself (except some standard or frequently used toolbox), and which is included in the Appendix chapter.

Chapter 2

Background Theory

This chapter is devoted to the theory behind the maximum entropy principle. In Section 2.1, the framework introduced in [7], including the primal problem, constraint qualification and the dual problem will be covered. Section 2.2 outlines the new problem now defined by new basis function, impossible price intervals and finally the duality problem. These two sections differ according to their basis functions: Section 2.1 uses the payoff function $c(x)$ and Section 2.2 uses the hat function $h(x)$.

To start the analysis, all the terms will be defined.

Definition 2.1. *A set X is said to be **convex** if*

$$\forall x, y \in X, \lambda \in [0,1] \rightarrow \lambda x + (1-\lambda)y \in X.$$

Definition 2.2. *A real valued function $f : X \rightarrow \mathbb{R}$ is said to be **convex** if*

$$\forall x, y \in X, \lambda \in [0,1] \rightarrow f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y).$$

*The function is said to be **strictly convex** if the above statement implies $f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda)f(y)$ instead of '$\leq$' when $\lambda \in (0,1)$.*

Definition 2.3. *Let S be a set in $\mathbb{R}^n$. The **convex hull** of S, denoted $conv(S)$ is the smallest convex set that contains S.*

Definition 2.4. *A subset M of $\mathbb{R}^n$ is called an **affine set** if*

$$(1-\lambda)x + \lambda y \in M \text{ for every } x \in M, y \in M \text{ and } \lambda \in \mathbb{R}.$$

Definition 2.5. *A set is called an **affine hull** of another set S, denoted $aff(S)$, in $\mathbb{R}^n$ if*

$$aff\ S = \left\{ \sum_{i=1}^{k} \alpha_i x_i \middle| k > 0, x_i \in S, \alpha_i \in \mathbb{R}, \sum_{i=1}^{k} \alpha_i = 1 \right\}.$$

Definition 2.6. *A set* $\boldsymbol{B}$ *in* $\mathbb{R}^n$ *defined with the usual Euclidean norm* $|\cdot|$ *is called the* ***Euclidean unit ball*** *if*

$$\boldsymbol{B} = \{x \big| |x| \leq 1\}$$

Definition 2.7. *For a set* $C \in \mathbb{R}^n$*, the* ***relative interior*** *of* C*,* $ri\ C$ *is defined as*

$$ri\ C = \left\{ \boldsymbol{x} \in \mathit{aff}\ C \middle| \exists \epsilon > 0, (\boldsymbol{x} + \epsilon \boldsymbol{B}) \cap \mathit{aff}(C) \subset C \right\}$$

Now we are ready to define our actual problem. Suppose there are m strike prices, $k_1, \ldots, k_m$, for an *European Call option* on a security X which has maturity at date T with corresponding observed market prices $d_1, \ldots, d_m$. The support for the strike prices is the interval $I = [0, K)$ where K is a bounded value (note that K can be infinite as well). We shall sort the strike prices in the way such that

$$0 = k_0 \leq k_1 \leq k_2 \leq \ldots \leq k_m \leq K.$$

The risk-neutral density $p(x)$ is the solution to the following problem:

$$\begin{aligned} &\text{find} \qquad \arg\min_p \quad \mathcal{I}_\phi(p) := \int_0^K p(x) \ln\left(p(x)\right) dx \\ &\text{subject to} \\ &\qquad 1 = d_0 = \int_0^K p(x)\, dx \\ &\qquad d_i = \int_0^K c_i(x) p(x)\, dx \qquad \text{for } i = 1, \ldots, m, \end{aligned} \tag{2.1}$$

(2.2)

where $c_i(x)$ are the option payoff function given by

$$c_i(x) = (x - k_i)_+ = \begin{cases} x - k_i & \text{if } x \geq k_i, x \in I \\ 0 & \text{otherwise.} \end{cases}$$

Note that in Equation (2.1), we have turned this into a minimising problem by taking the "negative" of the entropy functional we have in Equation (1.2) (we are only interested in the value of $p(x)$ but not $\mathcal{I}_\phi(p)$, so the sign does not matter here) (see [30], [21], [10]). Also we have chosen the Shannon entropy for $\phi(\cdot)$ where

$$\phi(t) = \begin{cases} t \ln t & \text{for } t > 0 \\ 0 & \text{for } t = 0 \\ +\infty & \text{for } t < 0. \end{cases}$$

2.1 Formulation under Payoff Function $c(x)$

2.1.1 Primal Problem

Define an operator $\mathbb{A} : L^1(I) \to \mathbb{R}^{m+1}$ for the constraints by

$$(\mathbb{A}p)_0 = \int_0^K p(x)\,dx$$
$$(\mathbb{A}p)_i = \int_0^K c_i(x)p(x)\,dx \qquad \text{for } 1,\ldots,m.$$

the problem defined in Equation (2.1) can be written as (see [8]):

$$\begin{aligned} &\arg\min_p \quad \mathcal{I}_\phi(p) := \int_0^K p(x)\ln\left(p(x)\right)dx \\ &\text{subject to} \\ &\qquad p \in L^1(I) \\ &\qquad \mathbb{A}p = \mathbf{d} = (1, d_1, \ldots, d_m)^T. \end{aligned} \tag{2.3}$$

2.1.2 Constraint Qualification

Borwein et. al [7] stated that the solution in (2.1) exists if and only if the data $\mathbf{d}$ lie within the feasibility region formed by the constraints. This condition is known as the "constraint qualification" (CQ) [7] which allows us to identify whether a solution exists before we actually run the optimisation. We will refer to this concept frequently throughout the thesis.

Definition 2.8. *The constraint qualification is:*

$$(CQ) \quad \boldsymbol{d} \in ri\ \mathcal{A} \tag{2.4}$$

where $\mathcal{A} := \{\boldsymbol{x} \in \mathbb{R}^{m+1} \mid \exists p \in L^1[0,\infty)$ *with* $\mathcal{I}_\phi(p)$ *finite and* $\mathbb{A}p = \boldsymbol{x}\}$.

Then $(1, d_1, \ldots, d_m)$ will satisfy the CQ (2.4) if and only if $(d_1, \ldots, d_m)^T$ satisfies

$$d_m > 0, \quad N^{-1}B_K(d_1,\ldots,d_m)^T > \mathbf{0} \quad \text{and} \quad \langle N^{-1}B_K(d_1,\ldots,d_m)^T, \mathbf{u}\rangle < 1 - \frac{d_m}{K-k_m} \tag{2.5}$$

where

$$N = \begin{bmatrix} k_2-k_1 & \cdots & k_m-k_1 \\ & \ddots & \vdots \\ & & k_m-k_{m-1} \end{bmatrix}, \qquad B_K = \begin{bmatrix} 1 & & -\frac{K-k_1}{K-k_m} \\ & \ddots & \vdots \\ & 1 & -\frac{K-k_{m-1}}{K-k_m} \end{bmatrix}$$

and $\mathbf{u}$ is a column vector of dimension $m-1$ whose elements are all equal to 1. This is the version of CQ for bounded K. Note that Equation (2.5) that Borwein et al. included the slightly erroneous form $\langle N^{-1}B_K(d_1,\ldots,d_m)^T,\mathbf{u}\rangle < 1$, where the correct form is corrected and added by Bose and Murray.

See [7] for the complete proof. What can be seen here is that the MES exists if and only if the data d lies in the region defined in (2.5), where the **open polyhedral set** is formed by the constraints.

2.1.3 Dual Problem

This section outlines the dual form of the primal problem (2.3).

First consider V and $V^\star$ to be two vector spaces with $\langle\cdot,\cdot\rangle$, a bilinear product on $V\times V^\star$, then the convex (Fenchel) conjugate of a convex function f on V is the function $f^\star$ defined on $V^\star$ by

$$f^\star(\xi) := \sup\left\{\langle x,\xi\rangle - f(x)\middle| x\in V\right\}. \tag{2.6}$$

Now in order to define the convex conjugate function $\Phi^*(p)$, we need the following Corollary by Rockafellar (see [27] and [8]).

Corollary 2.1.1. *Suppose that L and L^* are decomposable, and that T is of a finite measure. Let f be of the form $f(t,x)=F(x)$, where F is a lower semi-continuous proper convex function on $\mathbb{R}^n$. Then I_f on L and I_{f^*} on L^* are conjugate to each other.*

Then this shows that

$$[\mathcal{I}_\phi]^* = \mathcal{I}_{\phi^*}$$

The convex conjugate of a Shannon entropy is an exponential function, that is

$$\text{If } \phi(t)=t\log t, \text{ then } \phi^*(s)=e^{s-1}$$

Together, this led to the following result (see [7]):

$$\inf_{p\in L^1(I)}\left\{\mathcal{I}_\phi(p)\middle|\ p\in L^1(I),\mathbb{A}p=\mathbf{d}\right\} = \sup_{\lambda\in\mathbb{R}^{m+1}}\left\{\langle\lambda,\boldsymbol{\eta}\rangle - \mathcal{I}_\phi^\star(\mathbb{A}^T(\boldsymbol{\lambda}))\right\}. \tag{2.7}$$

This relation states that finding the $\boldsymbol{\lambda}$ that maximises the dual value function on the right hand side of (2.7) is equivalent of finding $p(x)$ which minimises the primal value function on the left hand side. The dual value function can be easier to solve since it is finite-dimensional.

Since the dual value function is one that

$$\text{find}\quad \max\left\{D(\lambda_0,\boldsymbol{\lambda}) := \lambda_0 + \sum_{i=1}^m \lambda_i d_i - \mathcal{I}\phi^\star\left(\mathbb{A}^T(\lambda_0,\boldsymbol{\lambda})\right)\ \middle|\ (\lambda_0,\boldsymbol{\lambda})\in\mathbb{R}^{m+1}\right\}$$

where $\mathbb{A}^T((\lambda_0, \boldsymbol{\lambda})) = \langle(\lambda_0, \boldsymbol{\lambda}), (1, \mathbf{c}(\cdot))\rangle$ is the adjoint of $\mathbb{A}$ such that $\mathbb{A}^T : \mathbb{R}^{m+1} \to L^\infty(I)$ and $[\mathcal{I}\phi]^\star\left(\mathbb{A}^T(\lambda_0, \boldsymbol{\lambda})\right)$ given by:

$$\begin{aligned}
[\mathcal{I}\phi]^\star\left(\mathbb{A}^T(\lambda_0, \boldsymbol{\lambda})\right) &= \int_I \phi^\star\left(\lambda_0 + \langle \boldsymbol{\lambda}, \mathbf{c}(x)\rangle\right) dx \\
&= e^{\lambda_0 - 1} \times \int_0^K e^{\sum_{i=1}^m \lambda_i (x-k_i)_+}\, dx \\
&= e^{\lambda_0 - 1} \times \sum_{i=1}^m \int_{k_j}^{k_{j+1}} e^{(\sum_{i=1}^j \lambda_i)x - \sum_{i=1}^j \lambda_i k_i}\, dx \\
&= e^{\lambda_0 - 1} \times \sum_{i=1}^m e^{-\nu_j} \frac{e^{\mu_j k_{j+1}} - e^{k_j \mu_j}}{\mu_j},
\end{aligned}$$

where $k_{m+1} := K$, $\nu_j := \sum_{i=1}^j \lambda_i k_i$, and $\mu_j := \sum_{i=1}^j \lambda_j$. Note that when $\mu_j = 0$, then $\frac{e^{\mu_j k_{j+1}} - e^{k_j \mu_j}}{\mu_j}$ is simply 0. Now if we redefine $e^{\lambda_0 - 1}$ as the normalisation factor of $p(x)$:

$$Z(\boldsymbol{\lambda}) := \int_0^K e^{\sum_{i=1}^m \lambda_i (x-k_i)_+}\ dx,$$

then the final dual function to be maximised is just

$$\ln Z(\boldsymbol{\lambda}) - \sum_{i=1}^m \lambda_i d_i. \tag{2.8}$$

and the maximum entropy solution (MES) is just $p(x) = \frac{1}{Z(\boldsymbol{\lambda}^\star)} e^{\sum_{i=1}^m \lambda_i^* c_i(x)}$ where λ^* maximises Equation (2.8).

If we compare this to the result in Equations (1.6) and (1.7), we see that they are exactly the same.

2.2 Formulation under $h(x)$

In [8], Bose and Murray used different basis function for the problem, which are the hat functions h. These hat functions have finite support (whereas the payoff functions do not) and allow one to spot the convexity violation of data through η, the moments of the hat functions, which will be mentioned later. Let us begin by defining these hat functions, $\{h_i(x)\}_{i=1}^{i=m+1}$, over the support $[0, K]$ such that the functions are linear interpolation of the following values:

$$h_1(k_j) = \begin{cases} 1 & \text{if } j = 0, 1, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{and for } i > 1 \quad h_i(k_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 2.1 shows an example of the hat functions:

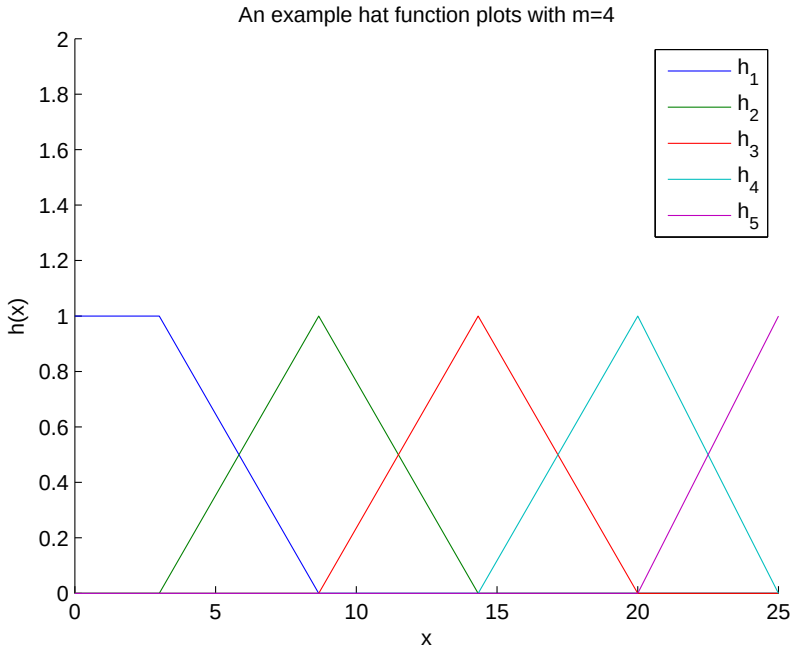

Figure 2.1: The hat functions $h_1, \ldots, h_m$ with $m = 4$.

From this, we can define $h(x)$ in terms of $c_i(x)$ and k_i, that is:

$$h_1 = 1 - \frac{c_1 - c_2}{k_2 - k_1}, \quad h_m = \frac{c_{m-1} - c_m}{k_m - k_{m-1}} - \frac{c_m}{K - k_m}, \quad h_{m+1} = \frac{c_m}{K - k_m},$$
$$\text{and} \quad h_i = \frac{c_{i-1} - c_i}{k_i - k_{i-1}} - \frac{c_i - c_{i+1}}{k_{i+1} - k_i} \quad \text{for } i = 2, \ldots, m-1.$$

Then we can write $h(x) = Bc(x)$ where c is the $(m+1)\times 1$ payoff vector and B is an $(m+1)\times(m+1)$ tridiagonal matrix with its non-zero terms defined in the following way (with $B_{i,j}$ specifying row i and column j of B):

$$\{B_{1,1}, B_{1,2}, B_{1,3}\} = \left\{1, \frac{-1}{k_2 - k_1}, \frac{1}{k_2 - k_1}\right\} \tag{2.9}$$

$$\{B_{i,i}, B_{i,i+1}, B_{i,i+2}\} = \left\{\frac{1}{k_i - k_{i-1}}, -\left(\frac{1}{k_i - k_{i-1}} + \frac{1}{k_{i+1} - k_i}\right), \frac{1}{k_{i+1} - k_i}\right\} \quad \text{, for } i = 2, \ldots, m-1$$

$$\{B_{m,m}, B_{m,m+1}\} = \left\{\frac{1}{k_m - k_{m-1}}, \left(\frac{1}{k_m - k_{m-1}} + \frac{1}{K - k_m}\right)\right\}$$

$$B_{m+1,m+1} = \frac{1}{K - k_m}$$

It is also easy to check that $\sum_{i=1}^{m+1} h_i = 1$

Let us introduce a moment called $\boldsymbol{\eta}$ by

$$\boldsymbol{\eta} = B\mathbf{d}$$

where B is the $(m+1) \times (m+1)$ matrix defined in Equation (2.9).

This vector $\boldsymbol{\eta}$ can be regarded as a version of our observed data $\mathbf{d}$ with the hat function transformation. To see this, since $h(x) = Bc(x)$, we can deduce from Equation (1.4) that $\boldsymbol{\eta} = \int_0^K h(x)p(x)\,dx$ which is the area under the density against the hat functions. The notion of $\boldsymbol{\eta}$ is inspired by [7] and first introduced by [8].

Recall the CQ that Borwein et al. raised in their paper in (2.5): for a maximum entropy solution to exist, the data d need to lie within the relative interior of the feasible polyhedral set. Bose and Murray's CQ is similar but constructed with $\boldsymbol{\eta}$ replacing d.

If the Equation (2.5) is expressed in terms of η, one can arrive at the following:

$$\eta_i > \mathbf{0} \quad \text{for } i = 2, \ldots, m$$

and

$$\sum_{i=2}^{m} \eta_i < 1 - \frac{d_m}{K - d_m}$$

Define $\eta_{m+1} := \frac{d_m}{K - d_m}$ and combine this with the other η_i to get $\sum_{i=2}^{m+1} \eta_i < 1$. Since the left hand side of this equation is a value less than, one can simply add another positive term, called η_1 to

make the inequality to an equality. That is

$$\begin{aligned} \text{If} \quad & \sum_{i=2}^{m+1} \eta_i < 1 \\ \text{then} \quad & \exists \eta_1 > 0 \quad s.t. \\ & \eta_1 + \sum_{i=2}^{m+1} \eta_i = 1 \\ & \sum_{i=1}^{m+1} \eta_i = 1 \end{aligned}$$

Note also that $\eta_1 = 1 - \frac{d_1 - d_2}{k_2 - k_1}$ because

$$\begin{aligned} \eta_1 &= 1 - \left(\sum_{i=2}^{m} \eta_i \right) - \eta_{m+1} \\ &= 1 - \left[\left(\frac{d_1 - d_2}{k_2 - k_1} - \frac{d_2 - d_3}{k_3 - k_2} \right) + \left(\frac{d_2 - d_3}{k_3 - k_2} - \frac{d_3 - d_4}{k_4 - k_3} \right) + \cdots + \left(\frac{d_{m-1} - d_m}{k_m - k_{m-1}} - \frac{d_m}{K - k_m} \right) \right] - \frac{d_m}{K - k_m} \end{aligned}$$

All the middle terms cancelled each other out and the result is clear.

Bose and Murray's CQ

The constraint qualification is satisfied if and only if $\boldsymbol{\eta}$ satisfies the following conditions:

$$\eta_i \geq \mathbf{0} \qquad \text{for } i = 1, \ldots, m+1, \tag{2.10}$$

and

$$\sum_{i=1}^{m+1} \eta_i = 1.$$

Using the same operator $\mathbb{B}$, by putting

$$\mathbb{B}p = B(\mathbb{A}p) = B\mathbf{d}$$

Then problem (2.3) becomes:

$$\begin{aligned} \min_{p} \quad & \mathcal{I}_\phi(p) := \int_0^K p(x) \ln\left(p(x)\right) dx \\ \text{subject to} \quad & \\ & p \in L^1(I) \\ & \mathbb{B}p = \boldsymbol{\eta}. \end{aligned} \tag{2.11}$$

Remarks:
The first condition in the (2.10) consists of $\boldsymbol{\eta} > \mathbf{0}$ the strict convexity condition or $\boldsymbol{\eta} = \mathbf{0}$ non-strict convexity condition. If $\eta_i > 0$, then MES exists satisfying the CQ with no problem. If η_i is zero, the CQ is still satisfied. But then the three data points (d_{i-1}, d_i, d_{i+1}) will be linear, which will lie on the boundary of the polyhedral set formed by these constraints. Bose and Murray suggested the interval (k_{i-1}, k_{i+1}) belongs to the impossible prices and should be removed from the interval $[0, K)$. If η_i is negative, then (d_{i-1}, d_i, d_{i+1}) are not convex and the condition $\boldsymbol{\eta} \geq \mathbf{0}$ in the CQ fails, and hence, the MES solution does not exist.

2.2.1 Impossible Prices

"Impossible prices" is a concept that Bose and Murray introduced in [8]. They claimed that if there exist intervals such that the corresponding η_i is zero then the interval should be removed from the support which they called this the "impossible price interval". As seen above if the data are not strictly convex, the solution does not exist. In order to solve a maximum entropy problem, $\boldsymbol{\eta}$ need to be all positive to satisfy the CQ.

Looking at the $\boldsymbol{\eta}$ in its integral form: $\eta_i = \int_0^K h_i(x)p(x)\,dx$ that given the non-negative hat functions and the non-negative risk-neutral density $p(x)$, the value of $\boldsymbol{\eta}$ will have to be non-negative. Negative

values in $\boldsymbol{\eta}$ suggest that the corresponding range of data points lie outside of the feasible polyhedral set. This implies some of the data are not convex and therefore violating only the Type A arbitrage-free principle. Hence we need to find appropriate strategies for preprocessing data to eliminate negative $\boldsymbol{\eta}$ values in order for the optimisation to work. If η_i is zero then the data lie on at least one of the boundaries of the polyhedral set, giving us a non-strict convexity over the region (k_{i-1}, k_{i+1}). We can see once again from the integral form of the η_i:

$$\eta_i = \int_0^K h_i(x)p(x)\,dx = \int_{k_{i-1}}^{k_{i+1}} h_i(x)p(x)\,dx.$$

If η_i equals zero, it is suggested either $h_i(x)$ is zero or $p(x)$ is zero in the region (k_{i-1}, k_{i+1}). In either case, $p(x)$ is essentially 0. Since $h_i(x) > 0$ on the interval (k_{i-1}, k_{i+1}), this suggests that $p(x)$ will be 0 in this interval. Because $p(x) = 0$ in the interval (k_{i-1}, k_{i+1}), it is impossible for prices that is consistent with the MES solution to exist in the interval.

Therefore **reduced domain** is definedI_0 as

$$I_0 = [0, K) - \bigcup_i (k_{i-1}, k_{i+1}) \qquad \text{for } i : \eta_i = 0.$$

From now on, the reduced domain I_0 is used instead of I for the integral since it is where the solution $p(x)$ lies. Because of this, a new operator, the **domain restricted operator**, $\mathbb{B}_0 : L^1(I_0) \to \mathbb{R}^{m+1}$ will replace $\mathbb{B}$.

We can now modify our primal problem noted in (2.11) as

$$\begin{aligned} \text{find} \quad & p := \arg\min \quad \mathcal{I}_\phi(p) := \int_{I_0} p(x)\ln\left(p(x)\right)dx \\ \text{subject to} \quad & \\ & p \in L^1(I) \\ & \mathbb{B}_0 p = \boldsymbol{\eta}. \end{aligned} \tag{2.12}$$

Let us define the adjoint operator of $\mathbb{B}_0 := \mathbb{B}_0^T : \mathbb{R}^{m+1} \to L^\infty(I_0)$ and

$$\mathbb{B}_0^T(\lambda) := \mathbf{1}_{I_0} \sum_{i=1}^{m+1} \lambda_i h_i \qquad \text{for each } \lambda \in \mathbb{R}^{m+1}.$$

Now assume that CQ holds. Then we can set out the Lagrangian to be:

$$\begin{aligned}
Q(\boldsymbol{\lambda}) &= \inf_{p} \left\{ \int_{I_0} \phi(p(x))\, dx + \sum_{i=1}^{m+1} \lambda_i \left(\eta_i - \int_{I_0} h_i(x) p(x)\, dx \right) \right\} \\
&= \inf_{p} \left\{ \Phi(p) + \langle \boldsymbol{\lambda}, \boldsymbol{\eta} - \mathbb{B}_0 p \rangle \right\} \\
&= \inf_{p} \left\{ \langle \boldsymbol{\lambda}, \boldsymbol{\eta} \rangle - \left(-\Phi(p) + \int_{I_0} \sum_{i=1}^{m+1} \lambda_i h_i p(x)\, dx \right) \right\} \\
&= \langle \boldsymbol{\lambda}, \boldsymbol{\eta} \rangle - \sup_{p} \left\{ \left(\int_{I_0} \mathbb{B}_0^T(\boldsymbol{\lambda}) p(x) d(x) - \Phi(p) \right) \right\} \\
&= \langle \boldsymbol{\lambda}, \boldsymbol{\eta} \rangle - \Phi^* \left(\mathbb{B}_0^T(\boldsymbol{\lambda}) \right)
\end{aligned}$$

where $f^\star(\xi) := \sup\left\{ \langle x, \xi \rangle - f(x) \middle| x \in V \right\}$ is the Fenchel conjugate defined in (2.6) with $f = \phi$; $x = p$; $\xi = \mathbb{B}_0^T(\boldsymbol{\lambda})$ and $f^\star = \phi^\star \in L^\infty(I_0)$.

Since the interval $[0, K)$ is finite, one can define a proper concave function $\phi(x)$. If $\Phi(f) = \int_0^K \phi(f(x))\, dx$, then by Corollary 2.1.1 its Fenchel conjugate can be defined on L^∞ as

$$\Phi^*(g) = \int_0^K \phi^*(g(x))\, dx$$

Recall that the convex conjugate of a Shannon entropy is just an exponential function. Therefore

$$\Phi^*(\mathbb{B}_0^T(\boldsymbol{\lambda})) = \int_0^K \phi^*(\mathbb{B}_0^T(\boldsymbol{\lambda}))\, dx = \int_0^K e^{\mathbb{B}_0^T(\boldsymbol{\lambda}) - 1}\, dx$$

So the dual value problem, Q, and the MES would be the $\boldsymbol{\lambda}$ that maximises Q

$$\max_{\lambda}\ Q = \langle \lambda, \boldsymbol{\eta} \rangle - \int_{I_0} e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1}\, dx \tag{2.13}$$

This can be achieved by solving the corresponding $\frac{\partial Q}{\partial \lambda_i} = 0$, where

$$\frac{\partial Q}{\partial \lambda_i} = \eta_i - \int_{I_0} h_i(x)\ e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1}\, dx \tag{2.14}$$

Our maximum entropy solution will be:

$$p(x) = e^{\sum_{i=1}^{m+1} \lambda_i^* h_i(x) - 1}$$

and this will be the desired risk-neutral density via the duality problem.

Chapter 3

Maximum Entropy Solution

This chapter is an experimental chapter with simulated data. It comprises of generating the simulated data adopted from Orozco-Rodriguez & Santosa's paper [26] and recovering the respective density, then followed by the generation of the random noise used in Chapter 4.

3.1 Log-Normal Prices

In order to explore an effective strategy for the noisy data, a benchmark true density is needed to compare the performance between the different strategies. This density needs to represent the market data well. The most frequently used density of underlying stocks is the log-normal. Not only was it the stock price density of the geometric Brownian motion that Black and Scholes used, it is also so commonly used in the study where results such as the formula of the partial expected value can be used.

Definition 3.1. *A continuous random variable X is distributed in the* ***log-normal distribution*** *with location parameter μ and scale parameter σ if its density function has the form*

$$p(x) = \frac{1}{x\sqrt{2\pi\sigma^2}}\, e^{-\frac{\ln x-\mu}{2\sigma^2}}.$$

This is also denoted as $\ln x \sim N(\mu, \sigma)$, where $N(\cdot)$ is the Normal distribution.

Suppose the market has m number of strike prices k ordered in the way that $0 \leq k_1 \leq k_2 \leq \cdots \leq k_m \leq k_{m+1} := K$. If k_i is fixed at some i, and by taking the expected value of the payoff function, one can easily obtain the log-normal option price pair $\{k_i, d_i\}_{i=1}^m$.

This can be shown easily as follow:

$$\begin{aligned} d_i = E_p[c_i] &= \int_0^K c_i(x)p(x)\,dx \\ &= \int_{k_i}^K (x-k_i)p(x)\,dx \\ &= \int_{k_i}^K xp(x)\,dx - k_i\int_{k_i}^K p(x)\,dx \\ &= \left(\int_{k_i}^\infty xp(x)\,dx - \int_K^\infty xp(x)\,dx\right) - k_i\left(\int_{k_i}^\infty p(x)\,dx - \int_K^\infty p(x)\,dx\right) \\ &= e^{\mu+\frac{1}{2}\sigma^2}\left[\Phi\left(\frac{\mu+\sigma^2-\ln k_i}{\sigma}\right) - \Phi\left(\frac{\mu+\sigma^2-\ln K}{\sigma}\right)\right] \\ &\quad - k_i\left[\Phi\left(\frac{\ln k_i-\mu}{\sigma}\right) - \Phi\left(\frac{\ln K-\mu}{\sigma}\right)\right]. \end{aligned} \tag{3.1}$$

where $\Phi(\cdot)$ is the Normal cumulative distribution function with parameters μ and σ.

Hence, the corresponding option price d_i can be readily obtained when strike prices k_i and parameters μ and σ have been specified. In [26], Orozco-Rodriguez and Santosa stated the log-normal density as

$$\ln x \sim N\left[\ln d_1 + \left(r - \frac{\sigma^2}{2}\right)T, \sigma^2 T\right]. \tag{3.2}$$

According to [19] and [26], the values of σ range from 0.15 to 0.60. Note that the σ in (3.2) is the volatility (not the same as the scale parameter defined earlier). For this reason, Orozco and Santosa picked two values to look at: volatility $\sigma = 0.5$ and 0.2. They also assumed a stock has a current price of $d_1 = 40$ dollars, maturity $T = 0.5$ years, expected rate of return $r = 0.16$. These parameters can then be readily substituted with $\mu = \ln d_1 + \left(r - \frac{\sigma^2}{2}\right)T$ and $\sigma = \sigma^2 T$ for the expected value calculation in (3.1). In their paper, they have specified d_i such that

$$d_i = e^{-rT}\int_0^\infty c_i(x)p(x)\,dx.$$

Note that their approach differs to the one listed in (3.1): first is the use of r. In this paper, the rate of return r, without loss of generality, is assumed to be zero to simplify calculation, so e^{-rT} is just 1. Second is the use of an unbounded strike price interval $(0,\infty)$, interval used in [8] is the general case $(0,K)$ where $K < \infty$ and $K = \infty$ both hold.

Next they defined four different sets of strike prices k for each of the volatility index:

When $\sigma = 0.5$,

$$\mathbf{k} = [0, 20, 40, 60, 80],$$
$$\mathbf{k} = [0, 10, 20, \ldots, 70, 80],$$
$$\mathbf{k} = [0, 5, 10, \ldots, 75, 80],$$
$$\mathbf{k} = [0, 4, 8, \ldots, 76, 80].$$

When $\sigma = 0.2$,

$$\mathbf{k} = [0, 25, 30, 40, 50, 60],$$
$$\mathbf{k} = [0, 25, 30, 35, \ldots, 55, 60],$$
$$\mathbf{k} = [0, 25, 27.5, 30, \ldots, 57.5, 60],$$
$$\mathbf{k} = [0, 25, 26, 28, 30, \ldots, 58, 60].$$

These data set **(k,d)** will be used throughout the chapter and whenever simulation is used.

Given the paired data set $(\mathbf{k}, \mathbf{d})$, recall that $\boldsymbol{\eta}$ can be calculated by

$$\boldsymbol{\eta} = B\mathbf{d}$$

Then the plots of the data $(\mathbf{k}, \mathbf{d})$ and the corresponding $\boldsymbol{\eta}$ are displayed in Figure 3.1 and 3.2 with $\sigma = 0.5$ and 0.2 respectively. The data sets have a strictly convex shape, which corresponds to the plots (b) since none of the $\boldsymbol{\eta}$ were negative.

Figure 3.1(a)

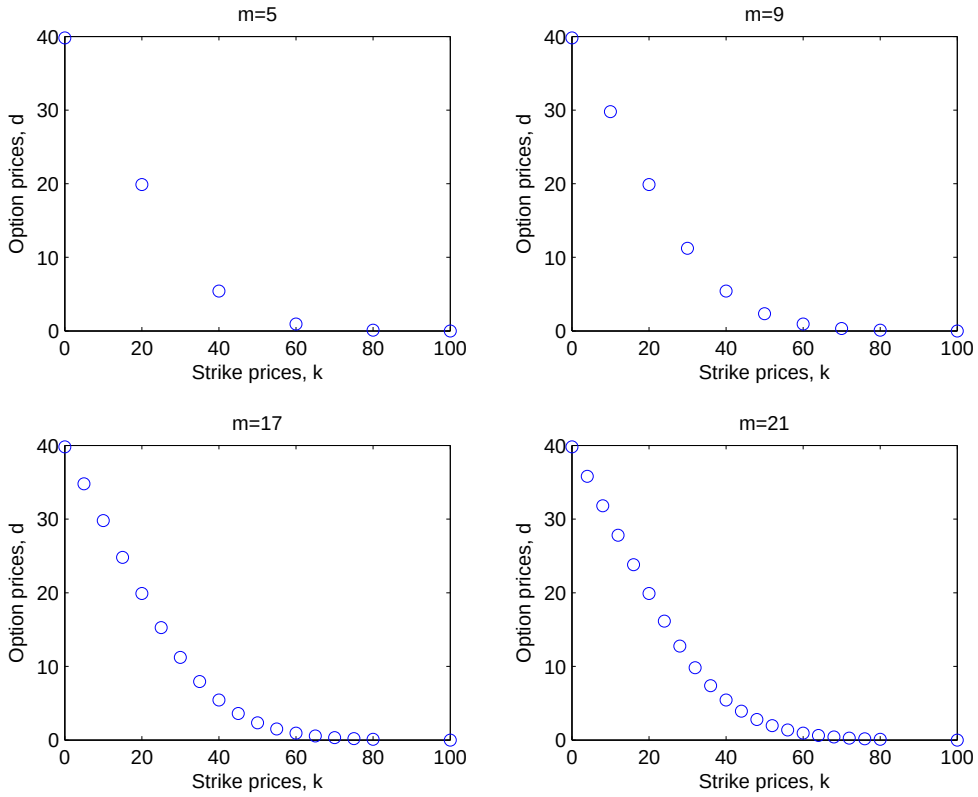

Figure 3.1(b)

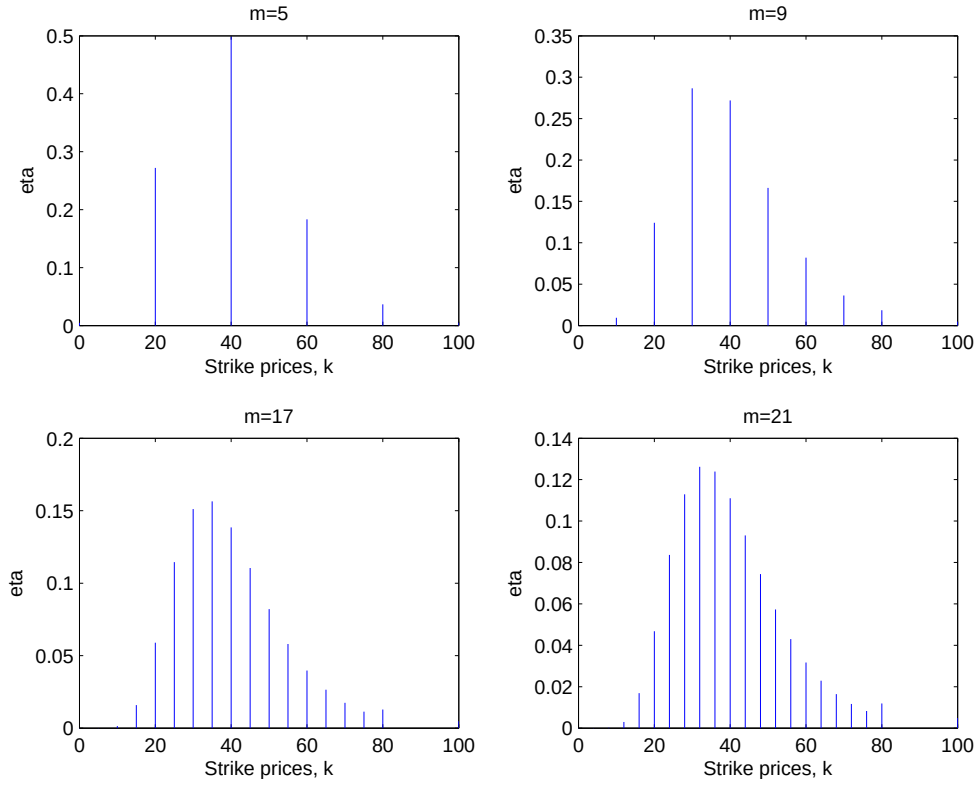

Figure 3.1: (a) plot of the data, (b) plot of $\boldsymbol{\eta}$ against strike prices $\mathbf{k}$ for $\sigma = 0.5$.

Figure 3.2(a)

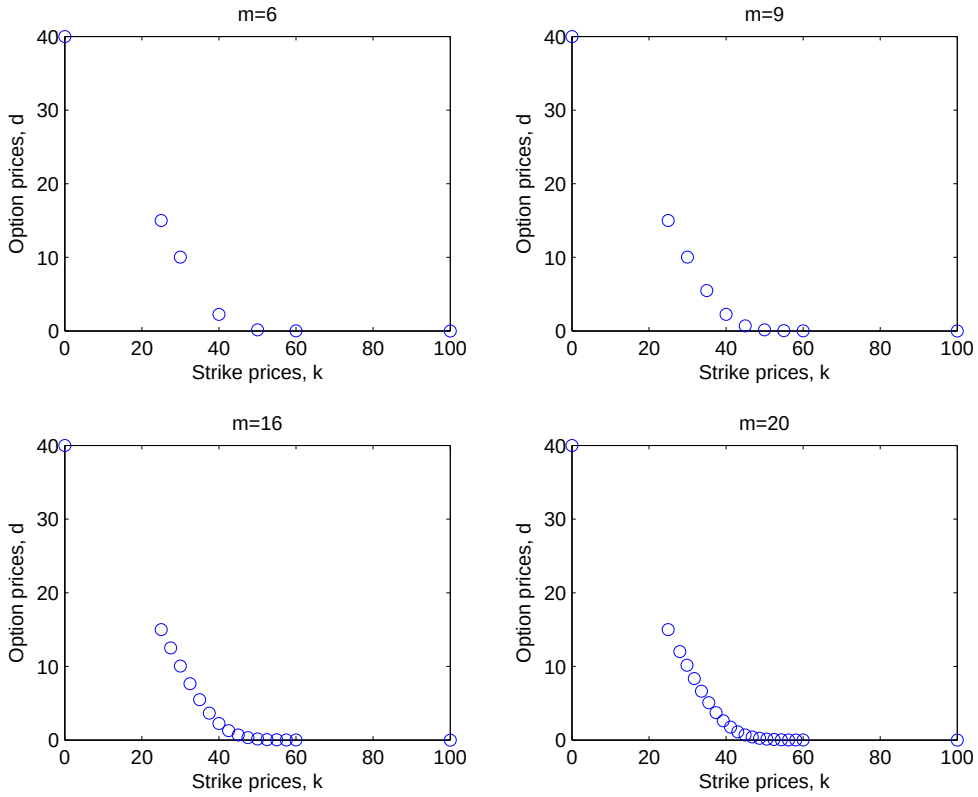

Figure 3.2(b)

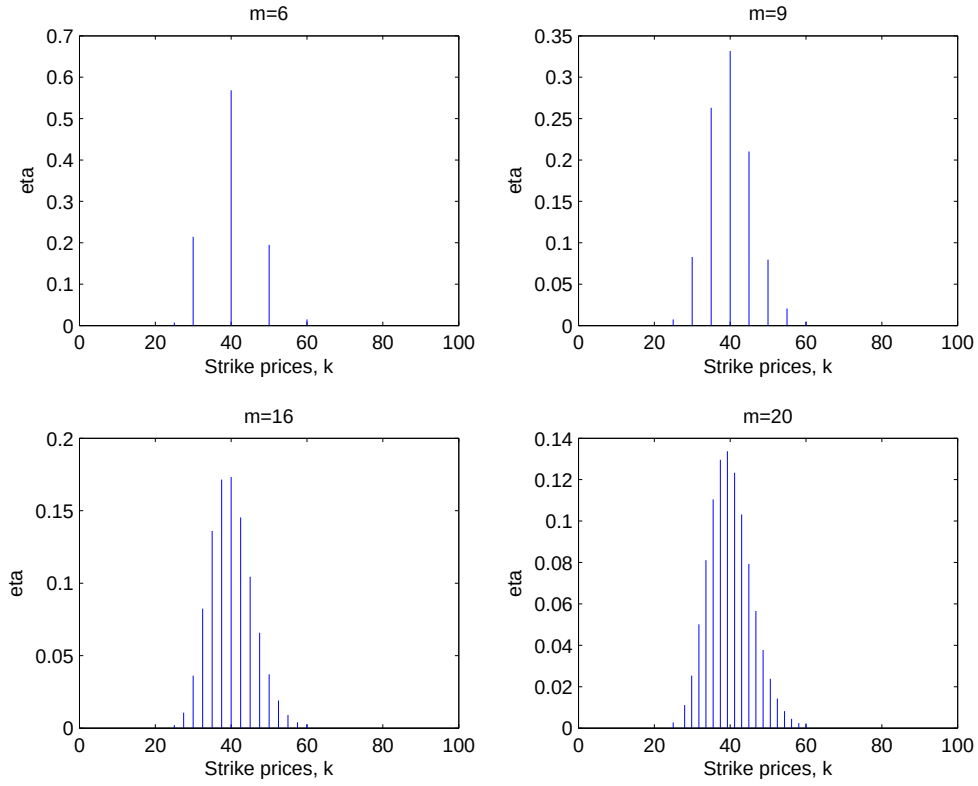

Figure 3.2: (a) plot of the data, (b) plot of $\boldsymbol{\eta}$ against strike prices $\mathbf{k}$ for $\sigma = 0.2$.

3.2 Density Reconstruction

The value problem Q and the optimality condition $\nabla_i Q$ in Equations (2.13) and (2.14) are as follow

$$Q = \langle \lambda, \boldsymbol{\eta} \rangle - \int_{I_0} e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1} \, dx,$$
$$\frac{\partial Q}{\partial \lambda_i} = \eta_i - \int_{I_0} h_i(x) \; e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1} \, dx.$$

One can easily deduce the Hessian matrix H by differentiating $\nabla_i Q$ with respect to λ_j as

$$H_{ij} = \nabla^2 Q = - \int_{I_0} h_i(x) \; h_j(x) \; e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1} \, dx.$$

Since the hat functions $h_i(x)$ are only positive in the interval (k_{i-1}, k_{i+1}) and zero elsewhere, the integrals of Q, $\nabla_i Q$, and H can be simplified to finite sum as follow

Q and $\nabla_i Q$:

$$Q = \langle \lambda, \eta \rangle - \sum_{i=1}^{m} (k_{i+1} - k_i) e^{\lambda_i - 1} \frac{e^{\lambda_{i+1} - \lambda_i} - 1}{\lambda_{i+1} - \lambda_i},$$
$$\frac{\partial Q}{\partial \lambda_i} = \eta_i - (k_i - k_{i-1}) e^{\lambda_{i-1} - 1} \frac{(\lambda_i - \lambda_{i-1}) e^{\lambda_i - \lambda_{i-1}} - e^{\lambda_i - \lambda_{i-1}} + 1}{(\lambda_i - \lambda_{i-1})^2}$$
$$- (k_{i+1} - k_i) e^{\lambda_i - 1} \frac{-(\lambda_{i+1} - \lambda_i) + e^{\lambda_{i+1} - \lambda_i} - 1}{(\lambda_{i+1} - \lambda_i)^2}.$$

The Hessian matrix, is a tridiagonal matrix with following entries:

$$H_{i,i} = -\Bigg[(k_i - k_{i-1})e^{\lambda_{i-1}-1}\frac{(\lambda_i-\lambda_{i-1})^2e^{\lambda_i-\lambda_{i-1}} - 2(\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} + 2e^{\lambda_i-\lambda_{i-1}} - 2}{(\lambda_i-\lambda_{i-1})^3}$$
$$+ (k_{i+1} - k_i)e^{\lambda_i-1}\frac{-(\lambda_{i+1}-\lambda_i)^2 - 2(\lambda_{i+1}-\lambda_i) + 2e^{\lambda_{i+1}-\lambda_i} - 2}{(\lambda_{i+1}-\lambda_i)^3}\Bigg] \quad \text{for } 2 \le i \le m,$$

$$H_{1,1} = (k_2 - k_1)e^{\lambda_1-1}\frac{-(\lambda_2-\lambda_1)^2 - 2(\lambda_2-\lambda_1) + 2e^{\lambda_2-\lambda_1} - 2}{(\lambda_2-\lambda_1)^3},$$

$$H_{m+1,m+1} = -\Bigg[(k_i - k_{i-1})e^{\lambda_{i-1}-1}\frac{(\lambda_i-\lambda_{i-1})^2e^{\lambda_i-\lambda_{i-1}} - 2(\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} + 2e^{\lambda_i-\lambda_{i-1}} - 2}{(\lambda_i-\lambda_{i-1})^3}$$
$$+ (k_{i+1} - k_i)e^{\lambda_i-1}\frac{-(\lambda_{i+1}-\lambda_i)^2 - 2(\lambda_{i+1}-\lambda_i) + 2e^{\lambda_{i+1}-\lambda_i} - 2}{(\lambda_{i+1}-\lambda_i)^3}\Bigg].$$

$$H_{i+1,i} = H_{i,i+1} = -(k_{i+1} - k_i)e^{\lambda_i-1}\frac{(\lambda_{i+1}-\lambda_i)e^{\lambda_{i+1}-\lambda_i} - 2e^{\lambda_{i+1}-\lambda_i} + (\lambda_{i+1}-\lambda_i) + 2}{(\lambda_{i+1}-\lambda_i)^3} \quad \text{for1} \le i \le m,$$

$$H_{i-1,i} = H_{i,i-1} = -(k_i - k_{i-1})e^{\lambda_{i-1}-1}\frac{(\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} - 2e^{\lambda_i-\lambda_{i-1}} + (\lambda_i-\lambda_{i-1}) + 2}{(\lambda_i-\lambda_{i-1})^3} \quad \text{for } 2 \le i \le m+1.$$

Note. only the cases of different λ are considered in this section ($\lambda_{i-1} \neq \lambda_i \neq \lambda_{i+1}$), where the other cases and the full derivation of these equations can be found in the Appendix (A.2).

Since the closed form of the Hessian matrix is easily implementable for programs in MATLAB, the fastest Newton's method was used for the optimisation routine with the stopping conditions for the step size and the gradient evaluated at each iteration both set to tolerance 10^{-12}.

The noiseless log-normal data set used here are convex and also satisfied the CQ, hence the solution λ^* converged very rapidly. The maximum entropy densities can then be obtained by

$$p^*(x) = e^{\sum_{i=1}^{m+1}\lambda_i^* h_i(x)-1}$$

The risk-neutral densities reconstructed are displayed in Figure 3.3(a) and (b) with $\sigma = 0.5$ and 0.2 respectively. The reconstruction outcome turned out to be very good as the reconstructed density have nearly perfectly matched onto the original log-normal density when $m = 21$ and 20 respectively.

Figure 3.3(a)

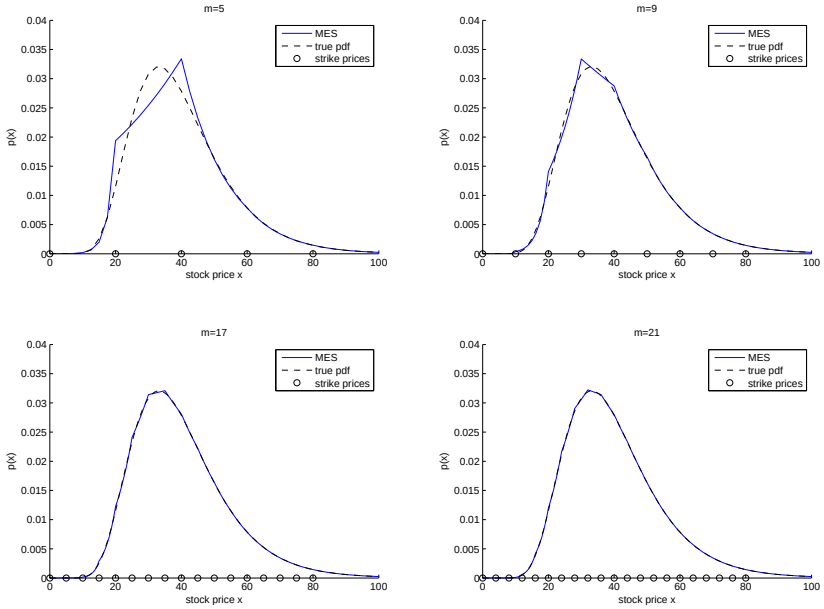

Figure 3.3(b)

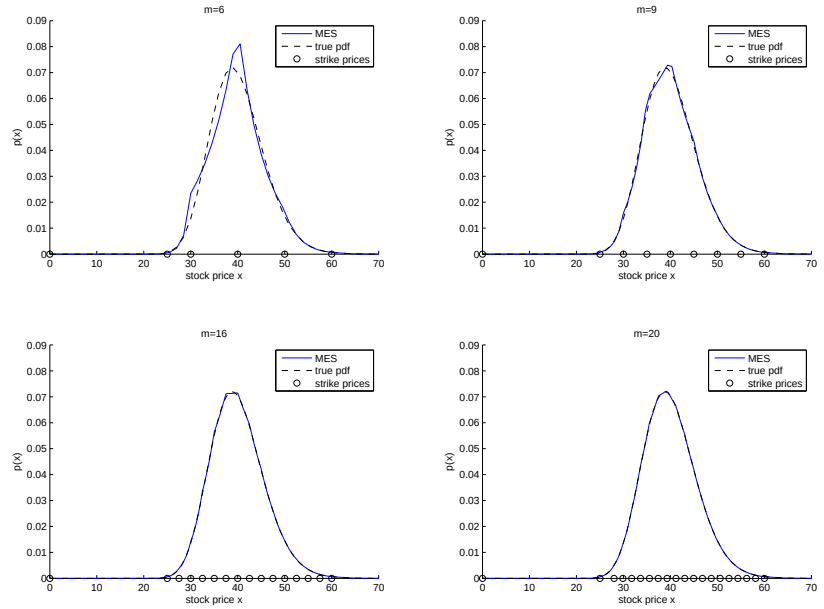

Figure 3.3: The maximum entropy solution of the reconstructed log-normal density for (a) $\sigma = 0.5$ and (b) $\sigma = 0.2$.

3.3 Performance

Integrated squared error is chosen to measure the performance of the density reconstruction such that the term is measuring the distance between two densities $p_{MES}(x)$ (maximum entropy density) and $p_{LN}(x)$ (log-normal density).

$$ISE = \int_{I_0} (p_{MES}(x) - p_{LN}(x))^2 \, dx.$$

The table of the ISE of the noiseless data is presented below

ISE	m_1	m_2	m_3	m_4
$\sigma = 0.5$	3.949×10^{-4}	2.861×10^{-5}	1.882×10^{-6}	7.752×10^{-7}
$\sigma = 0.2$	8.309×10^{-4}	5.421×10^{-5}	3.379×10^{-6}	1.096×10^{-7}

Table 3.1: The integrated squared errors of reconstructed log-normal density for various m (all rounded to 4 significant figures.)

Table 3.1 indicates that the higher the number of strike prices m were used in the density reconstruction, the estimation error gets smaller. Since it is totally arbitrary how the parameters were picked for the analysis, only the case of $\sigma = 0.5$ and $m = 21$ will be studied.

3.4 Simulating Noise

Recall the notion of CQ mentioned in Chapter 2. Since the data sets used above are all noiseless, all the corresponding η values were positive, it implies that MES exists (see also Figure 3.1 and Figure 3.2). However, if any of the η's are negative, then no solution will be found. In this section, noise will be added to the log-normal data set used earlier to explore some effective strategies that could remedy the situations when the data are non-convex or ill-conditioned.

The noisy data d_n is introduced by adding random white noise with a standard deviation that is proportional to the magnitude of the data point. Note that the noise was added only to $d \backslash \{d_1\} = \{d_2, \ldots, d_m\}$ since d_1 is the current spot price of the underlying stock with corresponding strike price of zero, it should be held fixed. Then the noisy piece of data set d_n is just

$$d_n = d + \epsilon.$$

where the ϵ is the white noise given by $\epsilon \sim N(0, \frac{d}{10})$. the factor $\frac{1}{10}$ was chosen so that the term $\frac{\|d - d_n\|_2}{\|d\|_2}$ can be kept at a level which is slightly less than 10% (when the number of replication gets larger, the mean of this ratio stayed at a level of 0.0960).

Figure 3.4 shows one realisation of the noisy data; the corresponding $\boldsymbol{\eta}$'s have changed significantly. some part of which has fallen below zero. This indicates the violation of CQ, so the solution of the maximum entropy problem will not be found. Hence, my main contribution to this thesis is to explore some useful existing or new strategies remedying this CQ violation and which are outlined in the next chapter.

Figure 3.4(a)

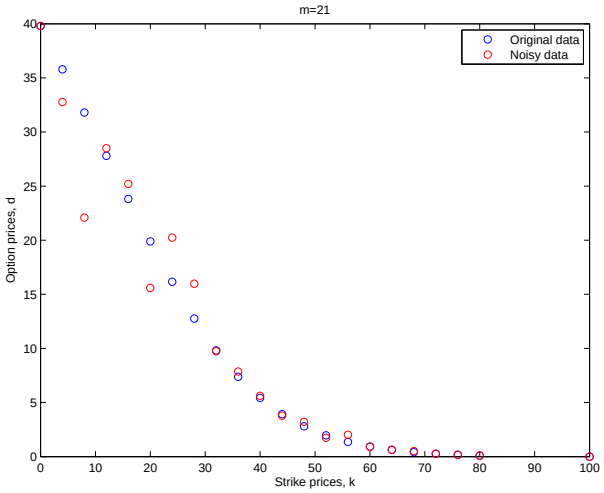

Figure 3.4(b)

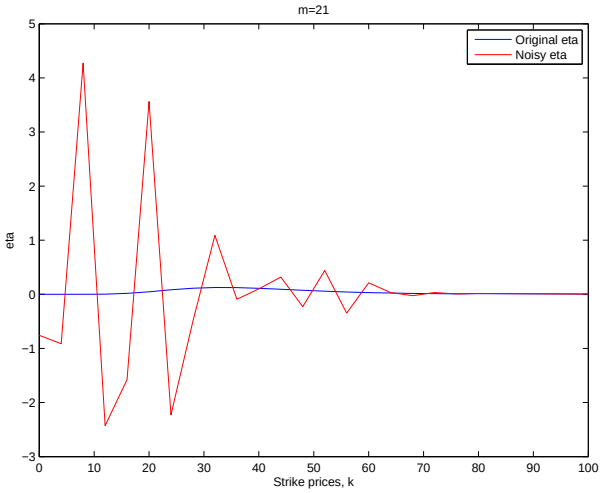

Figure 3.4: (a) plot of the noisy data, (b) plot of noisy $\boldsymbol{\eta}$ against strike prices $\mathbf{k}$ for $\sigma = 0.5$.

Chapter 4

Strategies for Noisy Data

Recall that CQ can fail when $\boldsymbol{\eta}$ has non-positive values. It was also shown that Bose and Murray's CQ and that of Borwein et. al are similar. The only difference is that Bose and Murray used one more constraint, that the sum of $\boldsymbol{\eta}$ needed to be one; because of this, the open polyhedral set was not open anymore. Furthermore, as a result of this condition, it required η_1 to be greater or equal to 0 in all cases.

In Chapter 2, $\boldsymbol{\eta}$ was defined by the matrix B such that $\eta_1 = 1 - \frac{d_1 - d_2}{k_2 - k_1}$. This is true only when $\mathbf{d}$ satisfies Borwein et. al's CQ, which states $\mathbf{d}$ needs to be in the relative interior of the operator $\mathcal{A}$. For a general data set $\mathbf{d}$, or simply a noisy data d_n, if the slope connecting d_1 and d_2 happened to be steeper than -1, the condition of $\eta_1 \geq 0$ would not stand as the term $\frac{d_1 - d_2}{k_2 - k_1}$ would be greater than 1.

A quick and immediate cure is to remove the condition $\sum_{i=1}^{m+1} \eta_i = 1$ and renormalise the density afterwards. This is achieved by simply removing h_1 from $\{h_i\}_{i=1}^{m+1}$, which effectively sets η_1 to be zero and removes the interval $(0, k_2)$ from the domain I_0. However, this will cause another problem: since h_2 needs to interpolate the three points $(k_2, 0)$, $(k_3, 1)$ and $(k_4, 0)$. This shows that $h_2(k_1)$ is now 0.

Recall that the MES has the form $p(x) = e^{\sum \lambda_i h_i(x) - 1}$. If the first hat function starts from 0, then $p(k_1) = e^{\sum \lambda_i h_i(k_1) - 1} = e^{-1} \approx 0.3679$ regardless of what the data is. This suggests that removing h_1 will not be a good idea.

One way that eventually works is to set η_1 to 0 but this would also change η_2 as a result. To see

this, recall the definitions of η_1 and η_2 as follow

$$\eta_1 = 1 - \frac{d_1 - d_2}{k_2 - k_1},$$
$$\text{and} \quad \eta_2 = \frac{d_1 - d_2}{k_2 - k_1} - \frac{d_2 - d_3}{k_3 - k_2}.$$

If η_1 is set to zero, this is essentially sets $\frac{d_1 - d_2}{k_2 - k_1}$ to 1. As a result, η_2 is now

$$\eta_2 = 1 - \frac{d_2 - d_3}{k_3 - k_2}.$$

Now observe the η_2 value, if it turns out to be negative again, this implies that the slope between d_2 and d_3 is again steeper than -1. If this is the case, set $\eta_2 = 0$ again and adjust $\eta_3 = 1 - \frac{d_3 - d_4}{k_4 - k_3}$. Repeat this process until first $\boldsymbol{\eta}$ is positive and halt there. At the end of this algorithm, the η_i being zeroed will trigger the corresponding (k_{i-1}, k_{i+1}) as impossible price intervals that will need to be removed from the domain I. One might be tempted to complete the full search of every negative η_i's in the vector $\boldsymbol{\eta}$ and set all of them to zero. That will then become a convex hull method since that is how the convex hull works. The whole idea of this method is to adjust the slope of the data so that they do not exceed -1, which is the steepest slope allowed in our problem.

This method of zeroing the initial η_i's is called the "data linearising method". Since what zeroing η_i really means is to assume there is a non-strict convexity, or linearity, between the data points. This method is actually a restricted convex hull method. This will be seen in Section 4.3 too. When discussing about other strategies in this chapter, the data linearising method will be applied whenever we observe the first few η_i's being negative, and this will be used along with the other strategies.

In the remaining sections, the experiments were replicated 1000 times where each time, the same log-normal data in Chapter 3 were used, along with randomly generated white noise added to it (will be referred to just $\mathbf{d}$). Then the integrated squared error of each simulation was recorded and compared with the other strategies. In each simulation, MES was used with Newton's method (as the closed form of Hessian matrix is known) and to increase precision.

4.1 Tikhonov Regularisation

This method was first applied by Orozco-Rodriguez and Santosa, specifying an extra penalty term to be added to the value function.

Tikhonov regularisation is a method often used to remedy an ill-conditioned optiomisation problem. This is achieved by adding a penalty term to the value function that is being optimised.

The value function in the previous section was given as:

$$Q = \boldsymbol{\eta}^T \lambda - \int_{I_0} e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1} \, dx$$

The regularised value function is

$$Q_\alpha(\lambda) = Q + \alpha \|\lambda\|^2.$$

Note that the penalty term $\alpha\|\lambda\|^2$ is added to the opposite direction to be optimised. In this case, the value function is being maximised, so the penalty term is subtracted from Q.

By selecting a value for α, the solution λ^*_α only solves the regularised dual problem, but not the original problem. Therefore, the densities reconstructed were normalised by

$$p(x) = \frac{e^{\sum_{i=1}^{m+1} \lambda^*_{\alpha_i} h_i(x) - 1}}{\int_{I_0} e^{\sum_{i=1}^{m+1} \lambda^*_{\alpha_i} h_i(x) - 1} \, dx},$$

where denominator value is just

$$\int_0^K e^{\sum_{i=1}^{m+1} \lambda^*_i h_i(x) - 1} \, dx = \sum_{i=1}^{m} (k_{i+1} - k_i) e^{\lambda^*_i - 1} \frac{e^{\lambda^*_{i+1} - \lambda^*_i - 1}}{\lambda^*_{i+1} - \lambda^*_i}.$$

as a result of cumulative distribution function in (A.1)

The values of α chosen on a trial-and-error basis were $\{0, 10^{-16}, 0.0001, 0.001, 0.01, 0.1, 0.25, 0.5, 1, 5\}$ (zero is included for comparison). Some selected reconstructed densities can be seen in Figure 4.1.

When α is smaller, the regularisation also has a smaller effect on remedying the Hessian - the boxplot in Figure 4.2 shows that there was no convergence for α less than 0.001. Recall that the spiky $\boldsymbol{\eta}$ in Figure 3.4(b) was caused by the presence of the noise in the data set, it can be seen that as α gets larger, the regularisation effect starts to kick in and the noise (shown as a result of the spikes in the densities) were flattened out until it approaches a uniform distribution. It can be seen in the mean integrated squared error plot (MISE) in Figure 4.2 that the 95% confidence intervals became smaller as α increased, this is due to an increase in the converged sample size (recall how the 95% confidence interval error bound is calculated by $z_{0.975} \frac{s}{\sqrt{n}}$). Hence, both the error bound and the confidence intervals were reduced when there were more simulations that converged.

Figure 4.1(a)

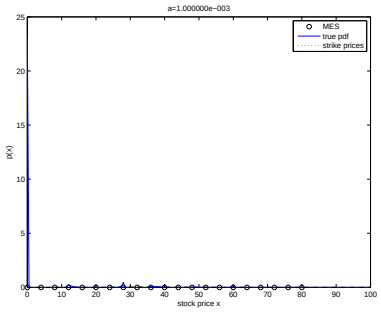

Figure 4.1(b)

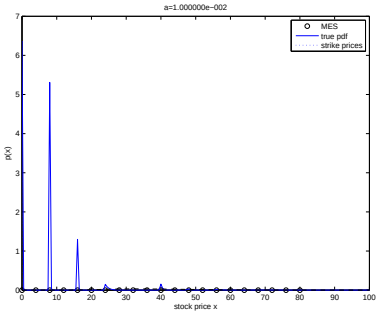

Figure 4.1(c)

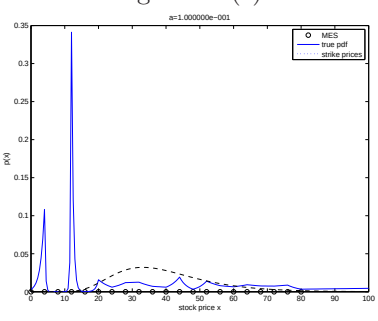

Figure 4.1(d)

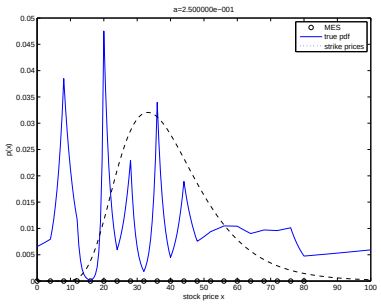

Figure 4.1(e)

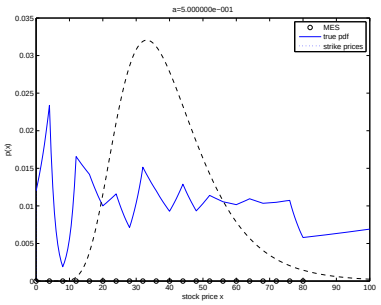

Figure 4.1(f)

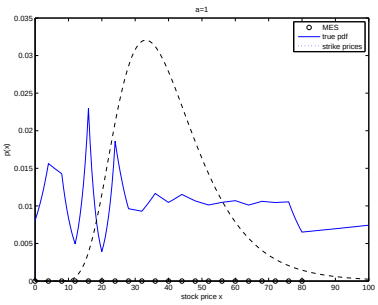

Figure 4.1(g)

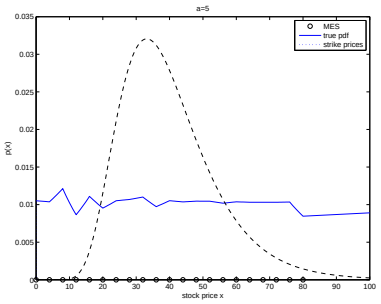

Figure 4.1: Some reconstructed density plots using the Tikhonov Regularisation method (a) $\alpha = 0.001$, (b) $\alpha = 0.01$, (c) $\alpha = 0.1$, (d) $\alpha = 0.25$, (e) $\alpha = 0.5$, (f) $\alpha = 1$, (g) $\alpha = 5$.

Both the integrated squared error (ISE) and the mean integrated squared error are at the minimum when $\alpha = 5$, giving a flatter distribution (and approaching a uniform distribution). A uniform distribution has not much use here since the MED has the same result as if there were no data set (Recall in Chapter 2). However, the proportion of convergence plot, in the log-scaled Figure 4.2(b), shows that every simulation converged (with proportion = 1) when α is greater than 0.01.

From this, it is evident that the regularisation method employed in [26] does not work well when the basis function is changed to hat functions $h(x)$. However, the best result obtained in this experiment is the one when $\alpha = 5$.

Figure 4.2(a)

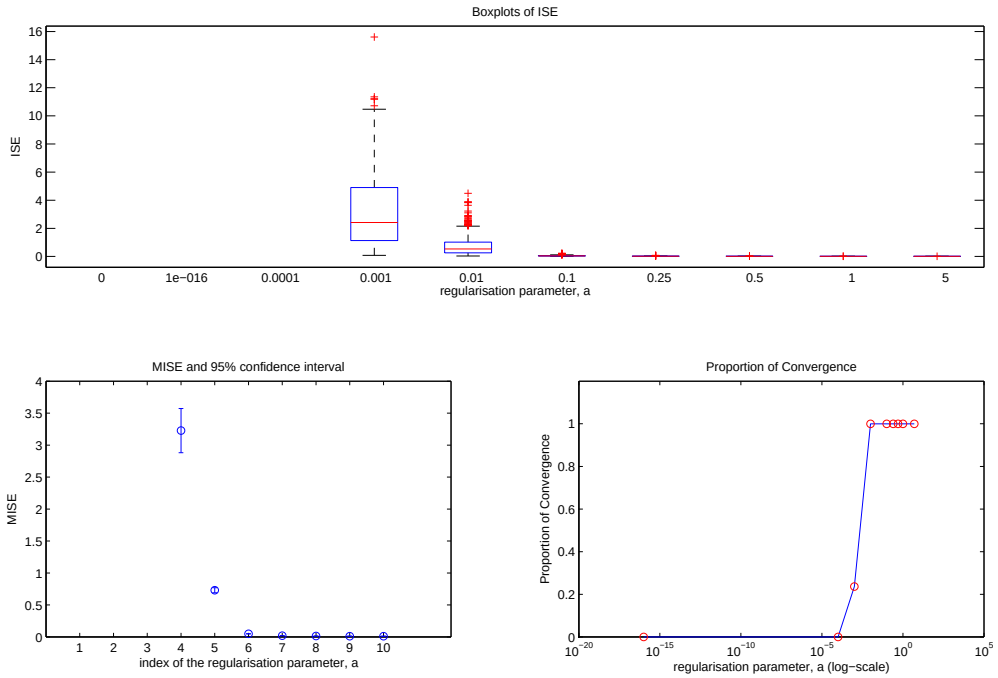

Figure 4.2(b)

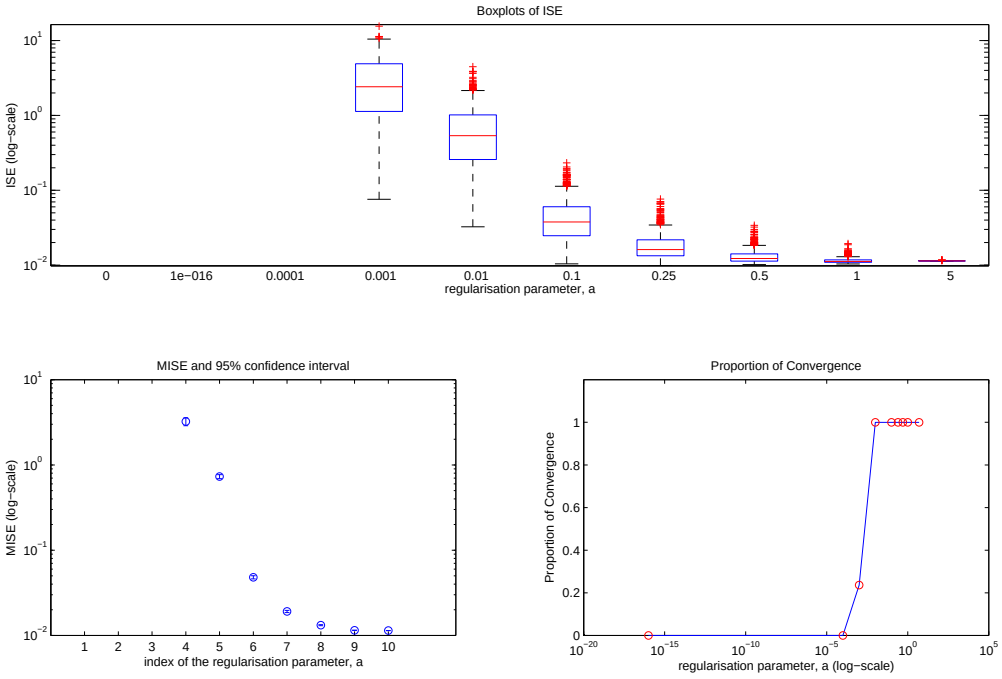

Figure 4.2: (a) The plots of the integrated squared error box plots, 95% confidence interval of mean integrated squared error, and the proportion of convergence via the Tikhonov regularisation method. The index of the α refers to the n-th element of the α where the values are shown at the x-axis of the box plots, (b) Log-scale of Figure 4.2(a).

4.2 Polyhedral Set Projection

This method, also introduced in [26], projects the data to the polyhedral set to have the CQ satisfied.

Recall Borwein et. al's CQ in (2.5) is

$$d_m > 0, \quad N^{-1}B_K(d_1,\ldots,d_m)^T > \mathbf{0} \quad \text{and} \quad \langle N^{-1}B_K(d_1,\ldots,d_m)^T, \mathbf{u}\rangle < 1 - \frac{d_m}{K-k_m}$$

where

$$N = \begin{bmatrix} k_2-k_1 & \cdots & k_m-k_1 \\ & \ddots & \vdots \\ & & k_m-k_{m-1} \end{bmatrix}, \qquad B_K = \begin{bmatrix} 1 & & -\frac{K-k_1}{K-k_m} \\ & \ddots & \vdots \\ & 1 & -\frac{K-k_{m-1}}{K-k_m} \end{bmatrix}$$

Then Orozco-Rodriguez and Santosa's projection problem is

$$\text{find} \quad \bar{\mathbf{d}} = \arg\min \|\bar{\mathbf{d}} - \mathbf{d}\| \tag{4.1}$$

$$\text{subject to} \tag{4.2}$$

$$N^{-1}B\bar{\mathbf{d}} \geq \epsilon\mathbf{u} \tag{4.3}$$

$$\mathbf{u}^T N^{-1}B\bar{\mathbf{d}} \leq 1-\epsilon \tag{4.4}$$

where ϵ is an arbitrary number, which they have chosen 10^{-4} in their examples.

Then $\bar{\mathbf{d}}$, which solves (4.1) is the projected data onto the open polyhedral set formed by the constraints.

After the basis function has been changed to the hat functions, these constraints and CQ were changed too. Recall the Bose & Murray's constraint qualification is given by:

$$\eta_i > 0 \quad \text{for } i = 1,\ldots,m+1$$
$$\sum_{i=1}^{m+1} \eta_i = 1$$

Notice these constraints now form a 'closed' polyhedral set in $\mathbb{R}^{m+1}$ space.

The projection optimisation problem is

$$\begin{aligned} \text{find} \quad & \bar{\mathbf{d}} = \arg\min \|\bar{\mathbf{d}} - \mathbf{d}\| \\ \text{subject to} \quad & \\ & \eta_i(\bar{\mathbf{d}}) > 0 \quad \text{for } i = 1, \ldots, m+1 \\ & \sum_{i=1}^{m+1} \eta_i(\bar{\mathbf{d}}) = 1 \end{aligned}$$

where $\boldsymbol{\eta}(\bar{\mathbf{d}}) = B * \bar{\mathbf{d}}$

Since $\mathbf{0}$ is on the boundary of the polyhedral set. This projection sets all the negative $\boldsymbol{\eta}$ values to be close to zero.

However, as $\boldsymbol{\eta}$ gets closer to zero the optimisation starts to break down (this is analysed in chapter 6). As a result of this, none of the 100 simulations had a solution. Therefore, the regularisation in section 4.1 was applied to this method. Figure 4.3 shows some of the density reconstructed.

These densities appeared to be very spiky and gets flatter as α increased (just as the case in Section 4.1). Figure 4.4 shows the box plots of the ISE and the MISE with 95% confidence interval. There were only few simulations converged when α is small (in the case when $\alpha = 0, 10^{-16}$ and both having 6 out of 1000 simulations converged). Hence, the corresponding box plots may not reflect an accurate measure of the ISE, where it has lower MISE than when α is 0.0001. The proportion of convergence hits one when α is greater than 0.0001. This shows how a tiny regularisation could remedy the convergence of the problem already.

This method gives an example that even when all elements of $\boldsymbol{\eta}$ were positive (after projecting to the polyhedral set satisfying the CQ), it does not guarantee a good fit in the MES reconstruction. While this method sets all negative η_i's to a small positive value close to zero, the remaining noise in the data (while their η_i were positive) were still present. This shows that even with a convex set of data, there is still a need to employ a method which deals with the problem of the noise itself. The cubic spline smoothing method is the one that smooths out the noise and is studied in Section 4.4. Before moving to that, another data-convexifying method is studied in the next section.

As a summary for the polyhedral set projection method, although it does not give us a very good fit (even with regularisation), $\alpha = 0.1$ appears to have the lowest MISE in Figure 4.4(b) and would be used to represent this method to compare with the other methods in the comparative analysis in Section 4.5.

Figure 4.3(a)

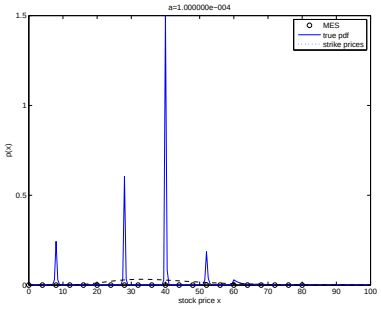

Figure 4.3(b)

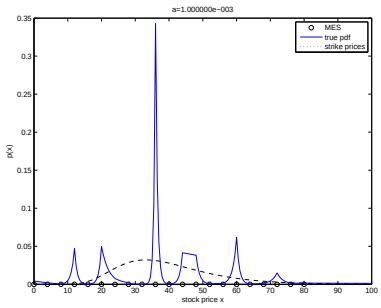

Figure 4.3(c)

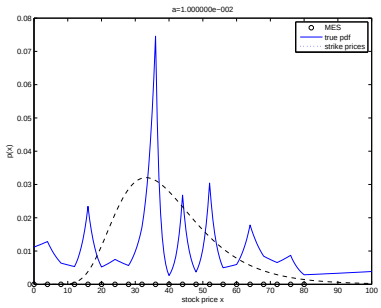

Figure 4.3(d)

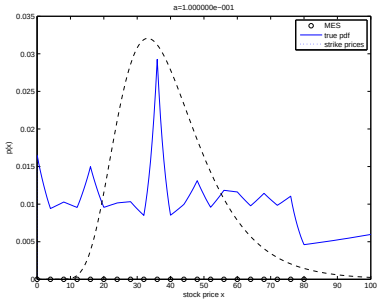

Figure 4.3(e)

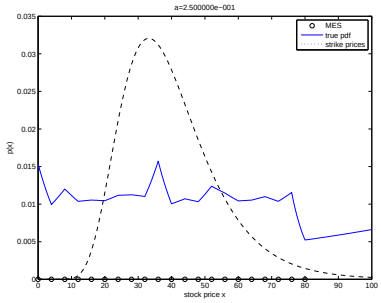

Figure 4.3(f)

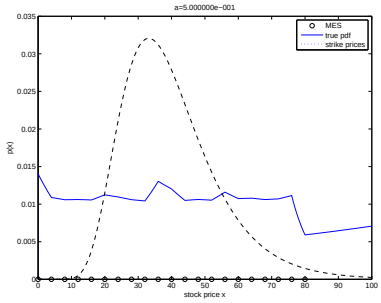

Figure 4.3(g)

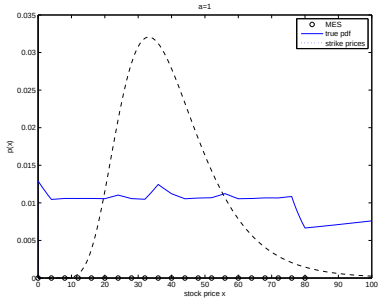

Figure 4.3(h)

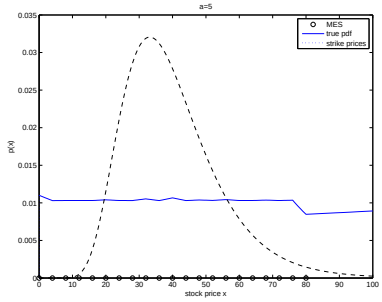

Figure 4.3: Some reconstructed density plots using the polyhedral set projection with regularisation (a) $\alpha = 0.0001$, (b) $\alpha = 0.001$, (c) $\alpha = 0.01$, (d) $\alpha = 0.1$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 1$, (h) $\alpha = 5$.

Figure 4.4(a)

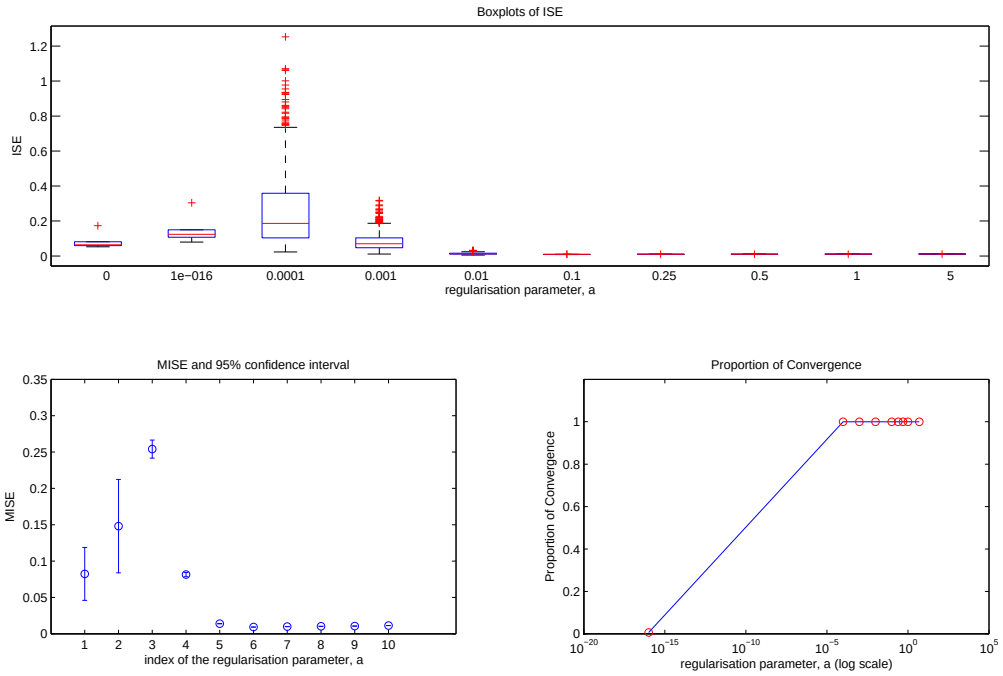

Figure 4.4(b)

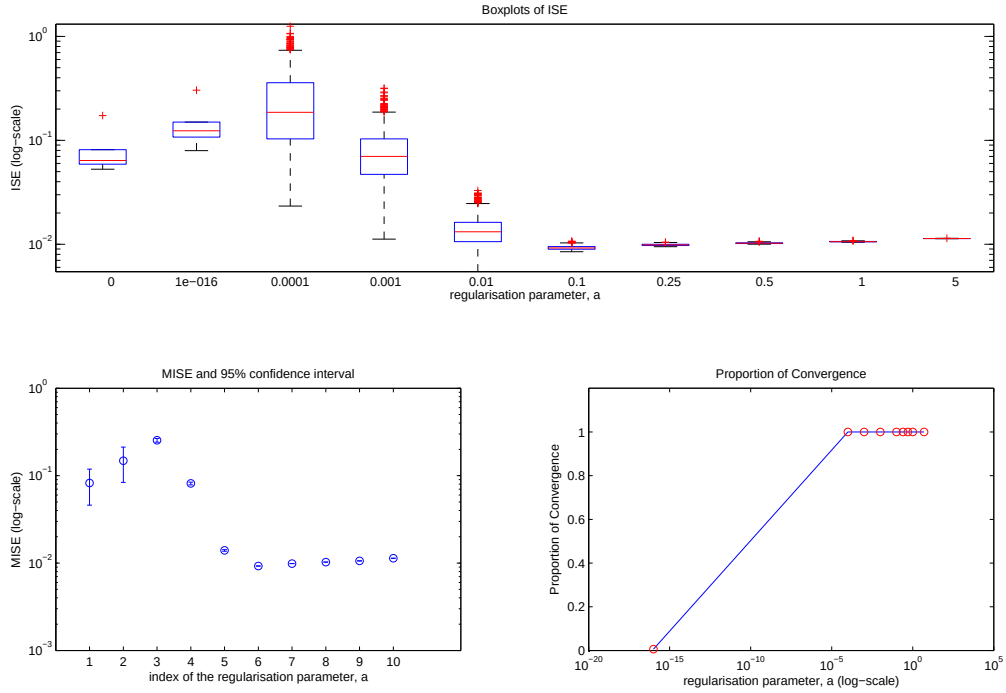

Figure 4.4: (a) The box plots of the integrated squared error and the 95% confidence interval of mean integrated squared error via the polyhedral set projection method. The index of the α refers to the n-th element of the α where the values are shown at the x-axis of the box plots, (b) Log-scale of Figure 4.4(a).

4.3 Convex Hull

When the data set is required to be convex at every point for CQ to hold, the most natural method is the convex hull method.

The convex hull of the data is one that has the first and end data point held fixed, so these two data points must be contained in the convex hull. Any point that does not lie on the boundary of the convex hull will then be removed.

This method is similar to the polyhedral set method and projecting the data not to an interior but the boundary of the set. That is:

$$\begin{aligned} \text{find} \quad & \bar{\mathbf{d}} = \arg\min \|\bar{\mathbf{d}} - \mathbf{d}\| \\ \text{subject to} \quad & \\ & \eta_i(\bar{\mathbf{d}}) \geq 0 \quad \text{for } i = 1, \ldots, m+1 \\ & \sum_{i=1}^{m+1} \eta_i(\bar{\mathbf{d}}) = 1 \end{aligned}$$

And then the region (k_{i-1}, k_{i+1}) where the corresponding η_i is zero will be removed from the domain. The removal of this is essentially similar to removing the data point that is not on the boundary of the convex hull.

Recall the data linearising method at the beginning of this chapter, mapping the first few negative η_i's to 0 are exactly the same as this method. The only difference is the linearising method stops when it encounters the first non-zero η_i.

Figure 4.5 shows some of the densities reconstructed with this method. The data linearising method discussed at the start of this chapter was used and the intervals corresponding to the zero η_i's were removed. This can be seen as part of the densities, including the original log-normal and MED, were removed and replaced with a vertical dash line. It can be seen that the first two $\alpha = 0, 10^{-16}$ values yield better results in the reconstructed density. As α increases, the problem becomes worse, as the regularisation method starts to destroy some of the key features of the original maximum entropy problem by making the density flat.

Figure 4.6 shows the performance of the convex hull method with various α values. The proportion of convergence plot also shows that although the convergence rate starts to break down as α gets bigger but the worst it gets is 98.8% which is pretty good. In Figure 4.6(b) specifically, the log-scaled box plots show that the first four α values have similar interquartile width, expressing most data are contained in similar range. As more regularisation added into the model, the variance of

Figure 4.5(a)

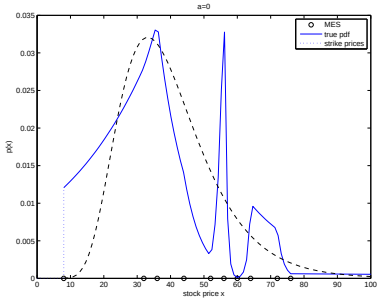

Figure 4.5(b)

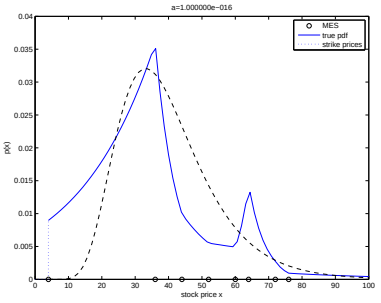

Figure 4.5(c)

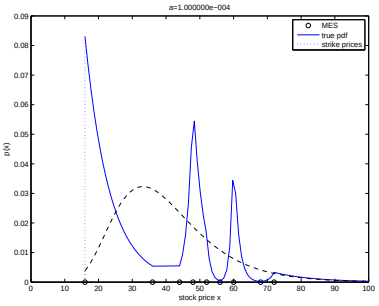

Figure 4.5(d)

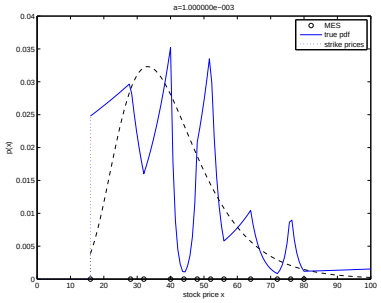

Figure 4.5(e)

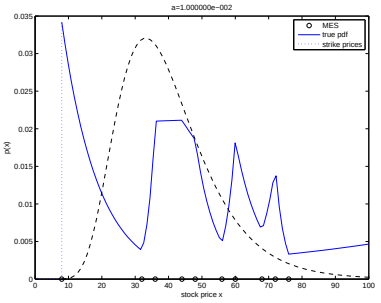

Figure 4.5(f)

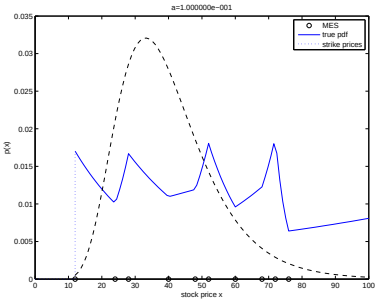

Figure 4.5(g)

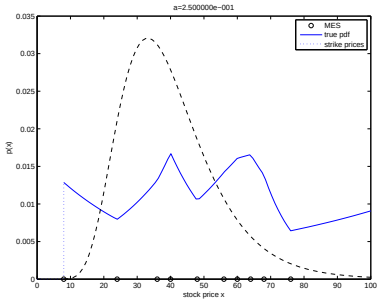

Figure 4.5(h)

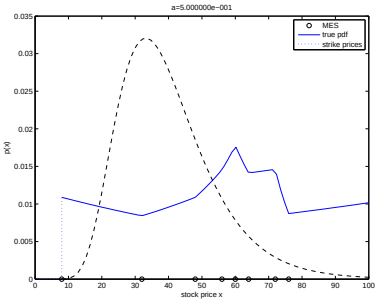

Figure 4.5(i)

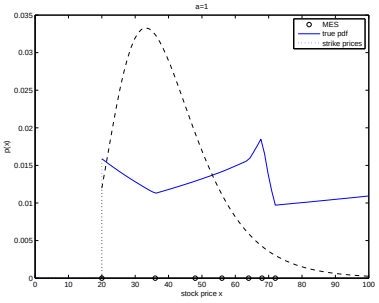

Figure 4.5: Some reconstructed density plots using the convex hull method with regularisation (a) $\alpha = 0$, (b) $\alpha = 10^{-16}$, (c) $\alpha = 0.0001$, (d) $\alpha = 0.001$, (e) $\alpha = 0.01$, (f) $\alpha = 0.1$ (g) $\alpha = 0.25$, (h) $\alpha = 0.5$, (i) $\alpha = 1$.

the ISE gets smaller, but also results in flatter distribution (approaching a uniform distribution).

The convex hull method gives visually good results when there is no or less regularisation. However, as α increases, it is shown that in Figure 4.6(b) that the MISE does not outperform the one with

more regularisations. As a result, the value ($\alpha = 0.1$) gives the lowest MISE, which will be used to represent the convex hull method in the model selection in Section 4.5.

Figure 4.6(a)

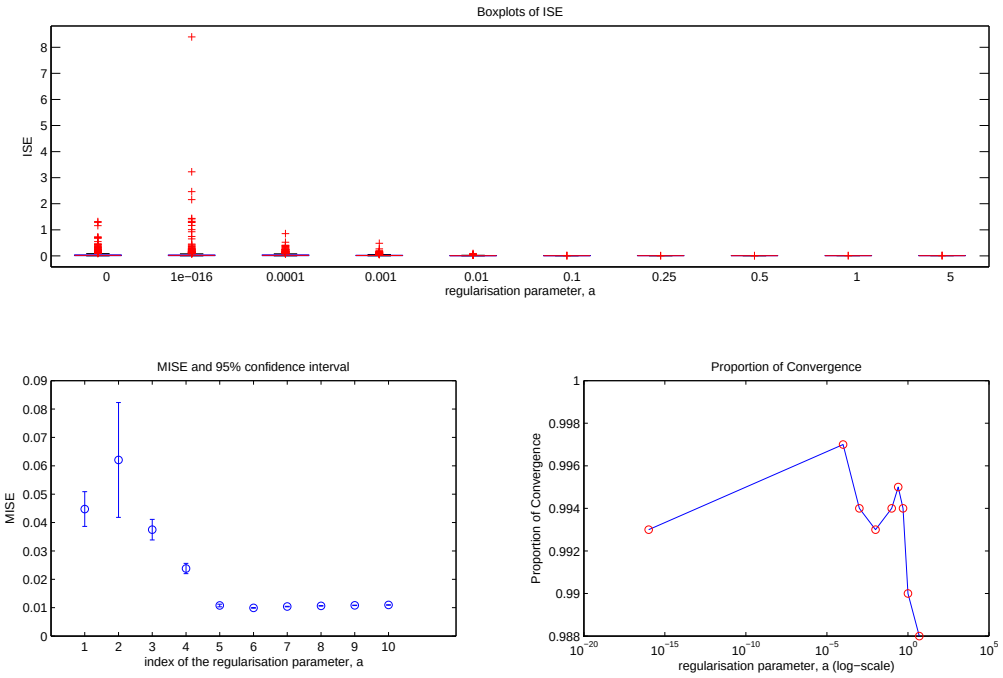

Figure 4.6(b)

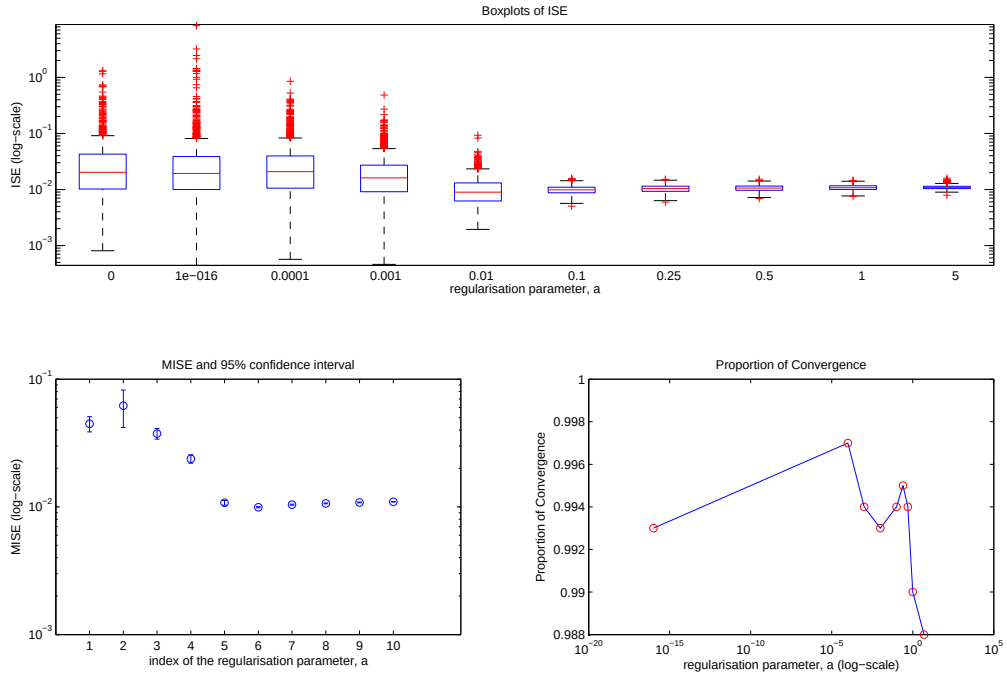

Figure 4.6: (a) The box plots of the integrated squared error and the 95% confidence interval of mean integrated squared error via the convex hull method. The index of the α refers to the n-th element of the α where the values are shown at the x-axis of the box plots, (b) Log-scale of Figure 4.6(a).

4.4 Cubic Spline Smoothing

Up to this stage, none of the strategies have addressed the noise reduction in the data set, but rather ignored and focussed on making the optimisation to work by adjusting the problematic data points or Hessian matrix. Cubic spline smoothing is one of the effective strategies for reducing noise. This method was inspired by Guo [15] and Monteiro et. al [24].

Cubic spline smoothing is a way to fit a data set that applies some smoothness while keeping the key features of the data. It exhibits a mixture of cubic spline interpolations and regression smoothing and is controlled by a parameter p.

Given a set of n points $(t_1, t_2, \ldots, t_n)$ and a data set $(y_1, y_2, \ldots, y_n)$, cubic spline spline smoothing finds a function f such that the following value function is minimised:

$$(1-p)\int (f''(t))^2 dt + p\,\frac{1}{n}\sum_{i=1}^{n}(y_i - f(t_i))^2 \tag{4.5}$$

Then the cubic spline smoothed (CSS) data set y_{CSS} is $\{f(t_i)\}_{i=1}^{n}$. In the case of option pricing, this can be rewritten to

$$(1-p)\int (f''(t))^2 dt + p\,\frac{1}{m-1}\sum_{i=2}^{m}(d_i - f(k_i))^2$$

Notice how the sum is ranging from 2 to m, this is because d_1 is fixed since it should be the spot price of the underlying stock.

Since the option prices d is bounded by zero, in order to ensure positive d is to transform it with log transformation where β_0 and β_1 are parameters and ϵ the random noise. Then d_{CSS} will just be $e^{f(k)}$. However, there exists bias in this transformation process.

Therefore, the bias corrected option prices in this problem is given as (the details are in the Appendix)

$$d_{CSS} = e^{f(k)}\ e^{\frac{\hat{\sigma}_{\lambda}^2}{2}}$$

The selection of the mixing parameter p is important too. As p approaches 1, then the cubic spline smoothing attempts to fit a cubic spline interpolation to the data whereas for p approaches 0, the regression smoothing kicks in. A range of 10 p values were chosen in this experiment where $p = \{10^{-8}, 10^{-4}, 2.5\times10^{-4}, 5\times10^{-4}, 10^{-3}, 2.5\times10^{-3}, 5\times10^{-3}, 0.01, 0.05, 0.75, 0.1, 0.4, 0.7, 1.000\}$.

Figure 4.7 shows the densities constructed by the cubic spline method. The reconstructions of this is by far the best of all strategies studied. There is also evidence for the impossible price interval in here too. It appeared that Figure 4.7(d) and (e) (where $p = 5 \times 10^{-4}, 10^{-3}$ respectively) have the best reconstruction as well, with most features of the original density recovered.

Figure 4.7(a) Figure 4.7(b) Figure 4.7(c)

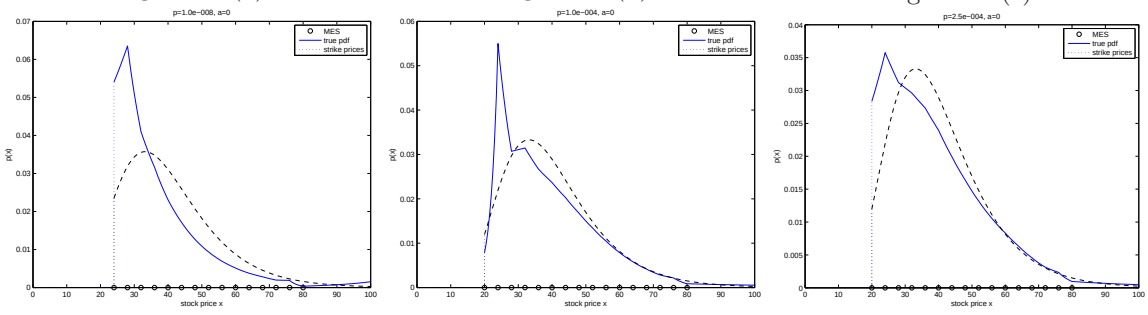

Figure 4.7(d) Figure 4.7(e) Figure 4.7(f)

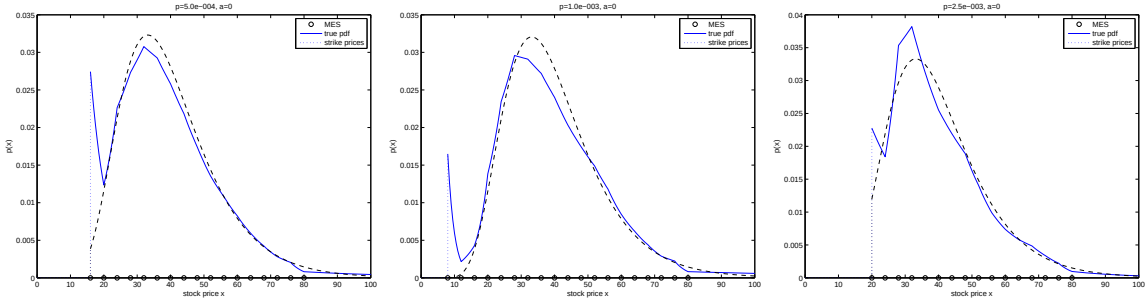

Figure 4.7(g) Figure 4.7(h) Figure 4.7(i)

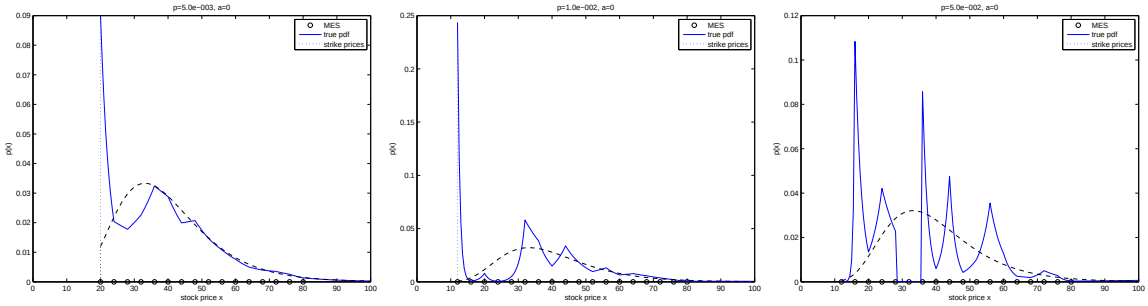

Figure 4.7: Some reconstructed density plots using the cubic spline smoothing method (a) $p = 10^{-8}$, (b) $p = 0.0001$, (c) $p = 0.00025$, (d) $p = 0.0005$, (e) $p = 0.001$, (f) $p = 0.0025$, (g) $p = 0.005$, (h) $p = 0.01$, (i) $p = 0.05$.

To see the whole picture of the 1000 replications, the box plots of the ISE and MISE are shown in Figure 4.8. It is shown that as p gets larger, the rate of convergence drops and the variance of the ISE gets bigger. Since a larger p favours the use of data more and less smoothing, more of the noise features are carried through the optimisation process, then it becomes more difficult for the data set to be convex and the optimisation to converge. From the log-scaled box plots of ISE in Figure 4.8(b), the variance of most values of p resemble one another. This is a good sign, since most of the densities reconstructed would then resemble one another in the same way, implying this method being effective and stable as long as the chosen p is not too large. The log-scaled MISE plots also reflect that $p = 5 \times 10^{-4}, 10^{-3}$ gives the best result, which corresponds to the Figure 4.7(d) and (e) earlier. Since the two mean integrated squared errors of the two p are similar, $p = 5 \times 10^{-4}$ is chosen as it appears to be slightly lower than the other.

Figure 4.8(a)

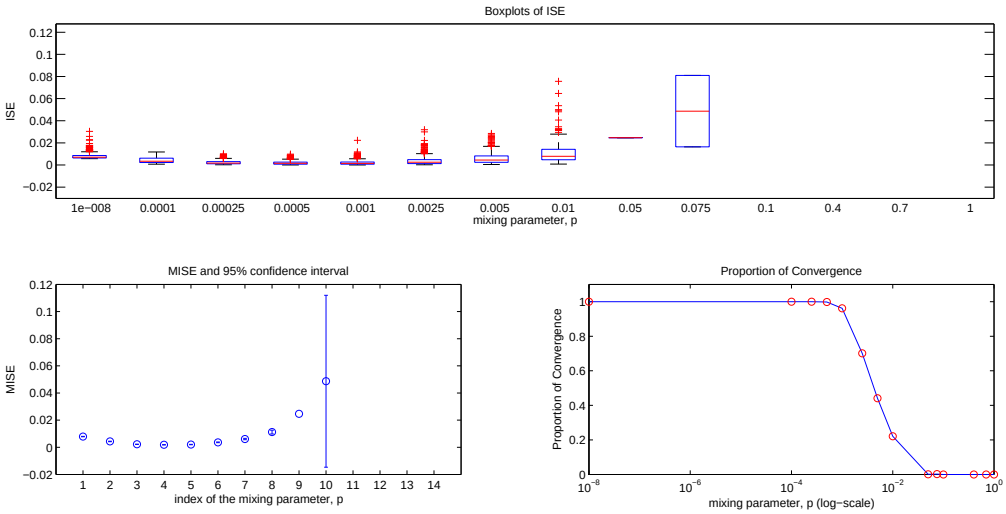

Figure 4.8(b)

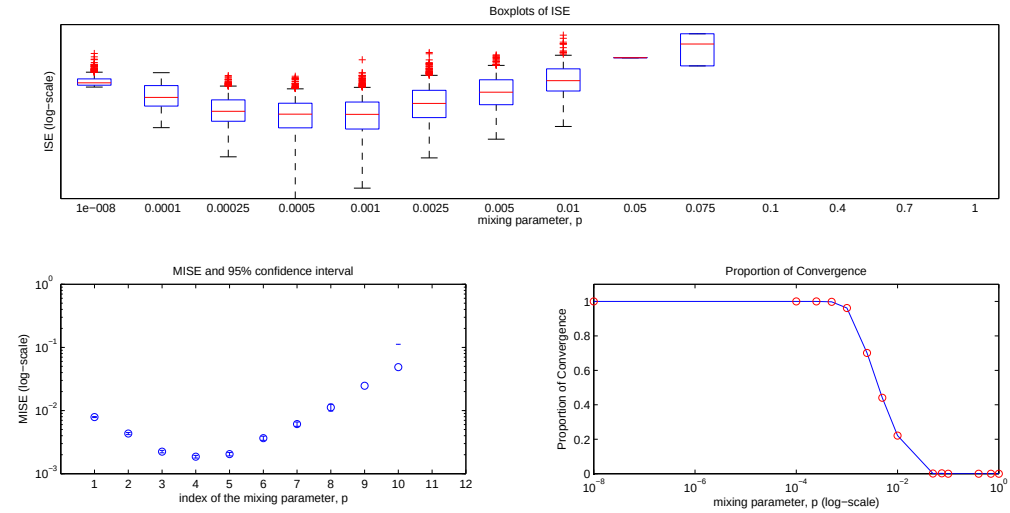

Figure 4.8: (a) The box plots of the integrated squared error and the 95% confidence interval of mean integrated squared error via the cubic spline smoothing method. The index of the parameter p refers to the n-th element of p where the values are shown at the x-axis of the box plots, (b) log-scaled plots of Figure 4.8(a)

4.5 Comparison of Different Approaches

From the four strategies presented: the Tikhonov regularisation, polyhedral set projection, convex hull method, and the cubic spline smoothing methods, one can compare which of these is preferred by pulling all of their integrated squared errors together. Recall in each strategy, there were certain parameters chosen such that the performance is better by picking the lowest integrated squared error amongst all the respective parameters. As a result, parameters $\alpha = 1$, $\alpha = 0.1$, $\alpha = 0.1$, $p = 0.0005$ were the ones representing their respective strategies. Figure 4.9 shows the respective densities once again. It is clear that Figure 4.9(d) is the densities that perform the best in that particular random simulation.

Figure 4.9(a)

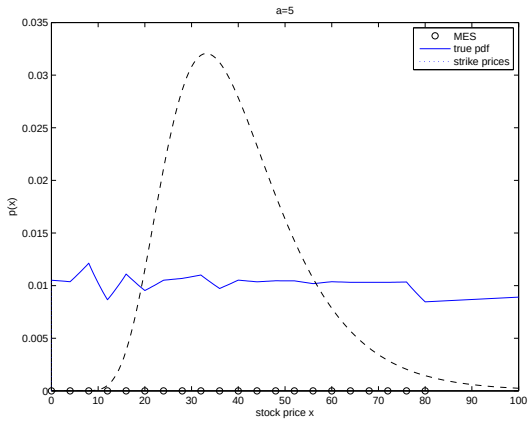

Figure 4.9(b)

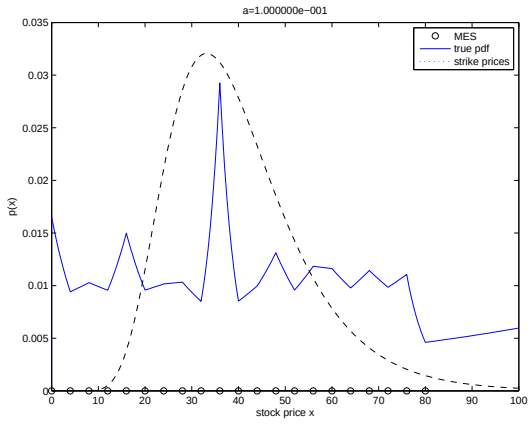

Figure 4.9(c)

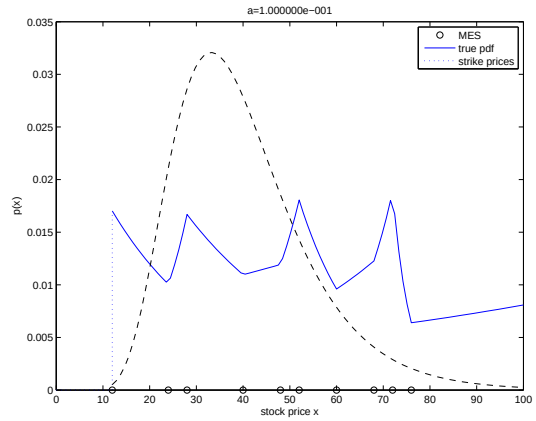

Figure 4.9(d)

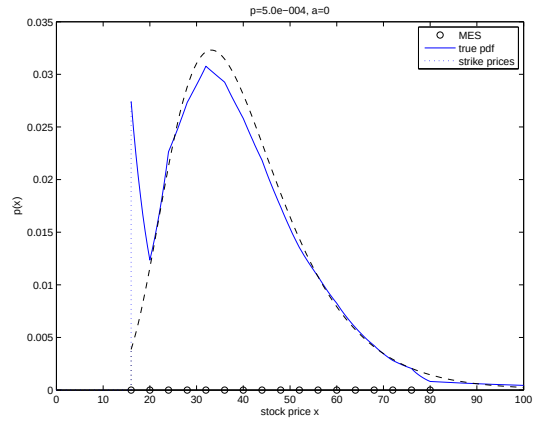

Figure 4.9: Reconstructed density plots of all four strategies (a) Tikhonov regularisation with $\alpha = 1$, (b) polyhedral set projection with $\alpha = 0.1$, (c) convex hull method with $\alpha = 0.1$, (d) cubic spline smoothing with $p = 0.0005$.

To see the whole picture of the 1000 replications, the box plots of their integrated squared errors

are presented in Figure 4.10. From the plot, it is shown that the variance of the Tikhonov regularisation is the smallest of all, implying across all replications the reconstructed densties are very similar (which totally makes sense as the problem is now overly regularised that a near-uniform distribution is obtained as a result). This is next followed by the polyhedral set projection, having slightly wider variance than the regularisation. Since the variance is not too large, the 1000 replications would look something like the density shown in Figure 4.9(b). The variance of the convex hull method and cubic spline smoothing methods are similar but larger than the previous two methods. This tells us that the densities generated with the two methods could vary relatively more than the previous two methods.

In terms of the levels of the four strategies, cubic spline smoothing has the lowest level in the ISE box plot and MISE plot, which implies it achieves the best out of all four strategies. Next followed by the polyhedral set projection method and convex hull method. Finally the regularisation has the highest ISE of all. Although this suggests that the cubic spline smoothing performs better than the others, one needs to bear in mind that this method also has relatively higher variance than the other strategies too.

In summary, it can be said that when noise was significant in the data, the best strategies is the cubic spline smoothing. This suggests thsy the best method to fit an MES onto a noisy data would be the one that would first reduce the level of noise and also allow the projected data sets be convex.

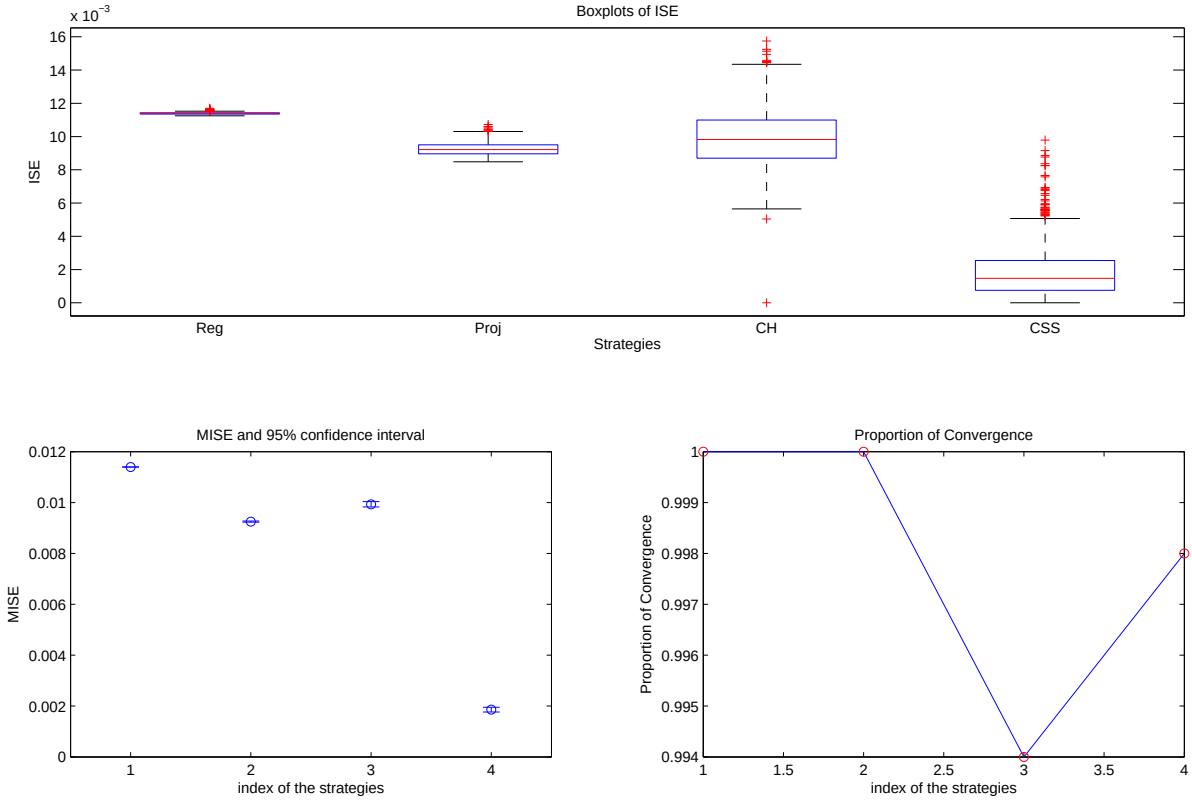

Figure 4.10: The integrated squared error box plots and the 95% confidence interval of mean integrated squared error of each strategy.

Chapter 5

Examples on Market Data VIX

From the last chapter, cubic spline smoothing has shown to outperform the other strategies. It is interesting to see how this method works on real data sets and this chapter will be completely exploratory. The data set used here is the Chicago Board Options Exchange (CBOE)'s volatility index of the S&P500 stocks (symbol: ˆVIX) on 15th December 2011. It was retrieved via Sirca in the Thomson Reuters Database (http://www.sirca.org.au/). Moreover, this data set consists of tick data, recording every trade and updating bid-ask price range with many tiny tick intervals.

Recall that in Chapter 4, one requires the first data point d_1 to be the current spot price of the security at that time. According to CBOE, the closing price of VIX on 15th December 2011 is \$25.11. Hence, a strike price of zero and \$25.11 will be used for the first data point (k_1, d_1).

Since this data set is a tick data set, there were so many data points available to be used in various ways. In order to reconstruct a risk-neutral density from the option prices, it is sufficient to pick only a single data point of option price for each of the different strike prices. There is a need to come up with a way choose which and how these data should be used.

This data set consists of 7 maturity dates and each maturity date has transaction occurred in almost every minute. For every strike price, the option prices with the same maturity were then grouped together and only one price data was chosen by finding their minimum, mean, median, weighted mean, and weighted median. In the end, the data set was organised into a 35×7 cell array where the row refers to the different maturity dates and the column refers to how the data was picked according to the 5 ways. Each array then consists of a set of m option prices, with one per each of the m strike prices. Consequently, the $\boldsymbol{\eta}$ of the data in each array could then be calculated and by taking the modulus (or the sum of the absolute value) of the $\boldsymbol{\eta}$, it is possible to identify the noisy level of each data set by observing how far its modulus of $\boldsymbol{\eta}$ is away from 1 (the sum of noiseless $\boldsymbol{\eta}$ would be 1). Then this allows the possibility of of selecting three data sets

where each was selected according to the data sets with minimum modulus, median modulus and maximum modulus of $\boldsymbol{\eta}$ out of all data in the 35×7 cell array data sets. These three data sets were studied by applying the cubic smoothing spline method and MES could then be obtained. Figure 5.1 shows the plots of the data points and $\boldsymbol{\eta}$ of the three data sets chosen. It is immediate that the convexity is violated when some of the η_i's were negative as all three of them have modulus of $\boldsymbol{\eta}$ greater than 1. The maximum modulus being 4.04 in 5.1(c) is also quite far from 1.

Figure 5.1(a)

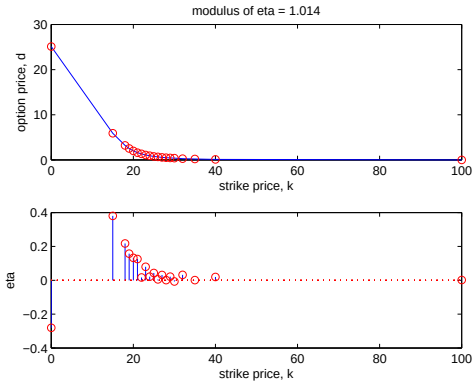

Figure 5.1(b)

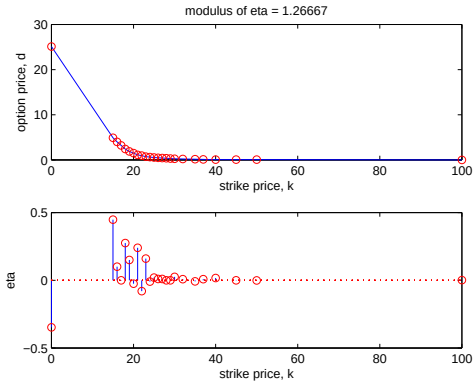

Figure 5.1(c)

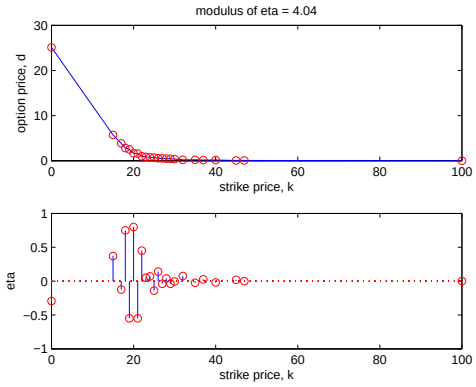

Figure 5.1: Plots of the three data sets and their $\boldsymbol{\eta}$'s with (a) data set with minimum modulus of $\boldsymbol{\eta}$, (b) data set with median modulus of $\boldsymbol{\eta}$, (c) data set with maximum modulus of $\boldsymbol{\eta}$.

Since it is not possible to refer to an absolute true density to measure the performance of the strategy when it comes to real data. The approach taken here is to attempt a range of parameters p such that densities could be reconstructed, and working back from the highest end of p and accept the density which looks visually fine. This is because a higher p actually makes more use of the data points available (recall that higher the p, the scheme favours data point interpolation whereas lower p favours smoothing).

5.1 Data with Minimum Modulus of η

Figure 5.2 shows the reconstructed densities with a range of p. Although there is convergence with such a high p value of 0.9, the density seems to be a little bit spiky on some of the edges, the next smoothest densities would have to be when $p = 0.1$ when the spikes in the tails and the middle have disappeared.

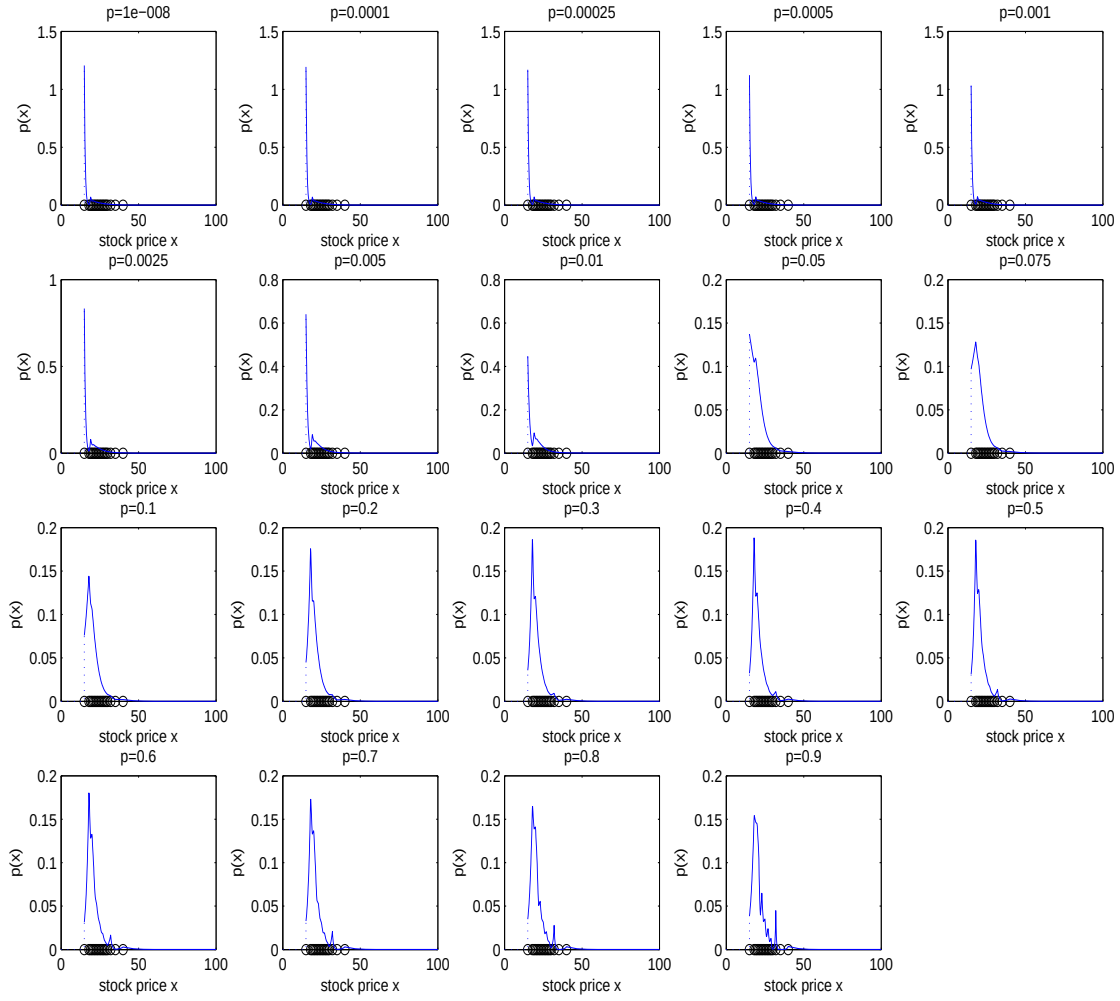

Figure 5.2: The reconstructed cubic spline smoothing density of VIX option index data corresponding to the minimum modulus of η

5.2 Data with Median Modulus of η

Figure 5.3 indicates there were fewer convergence compared to the data set with minimum modulus, this is because there were either more or larger negative η_i's in the data now. Starting from $p = 0.01$, there is a very large spike appearing in the density where it is likely to have occured from the noise in the data. The densities presented here were quite similar to each other, perhaps the density with $p = 0.01$ performed a little better.

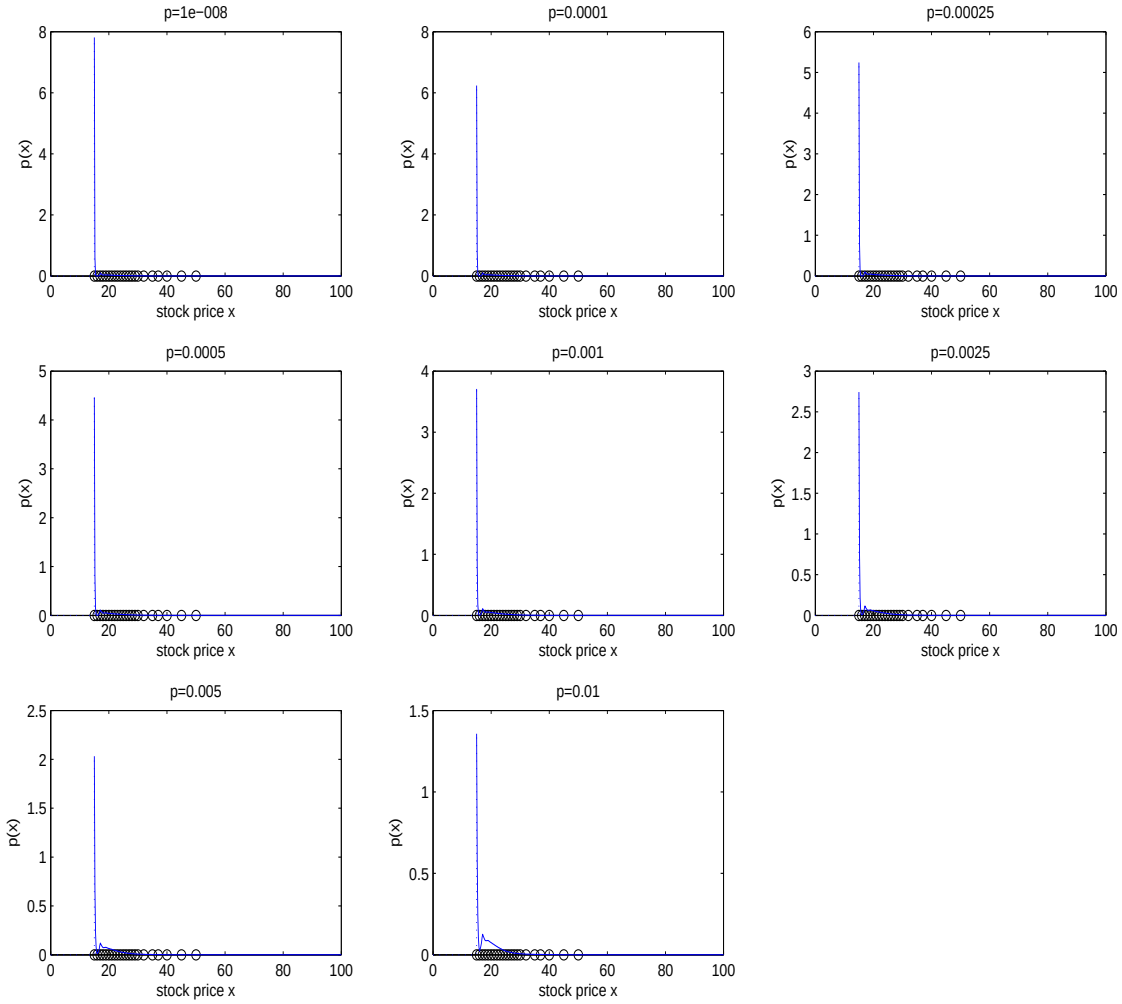

Figure 5.3: The reconstructed cubic spline smoothing density of VIX option index data corresponding to the median modulus of $\boldsymbol{\eta}$

5.3 Data with Maximum Modulus of η

When fitting the MES onto the data set corresponding to the maximum modulus of $\boldsymbol{\eta}$, more negative $\boldsymbol{\eta}$ appeared and resulted in fewer convergence with the same range of p values. The first convergence occurred when $p = 0.05$, where it happened to be the best density recovered which is both smooth and no obvious spikes were present.

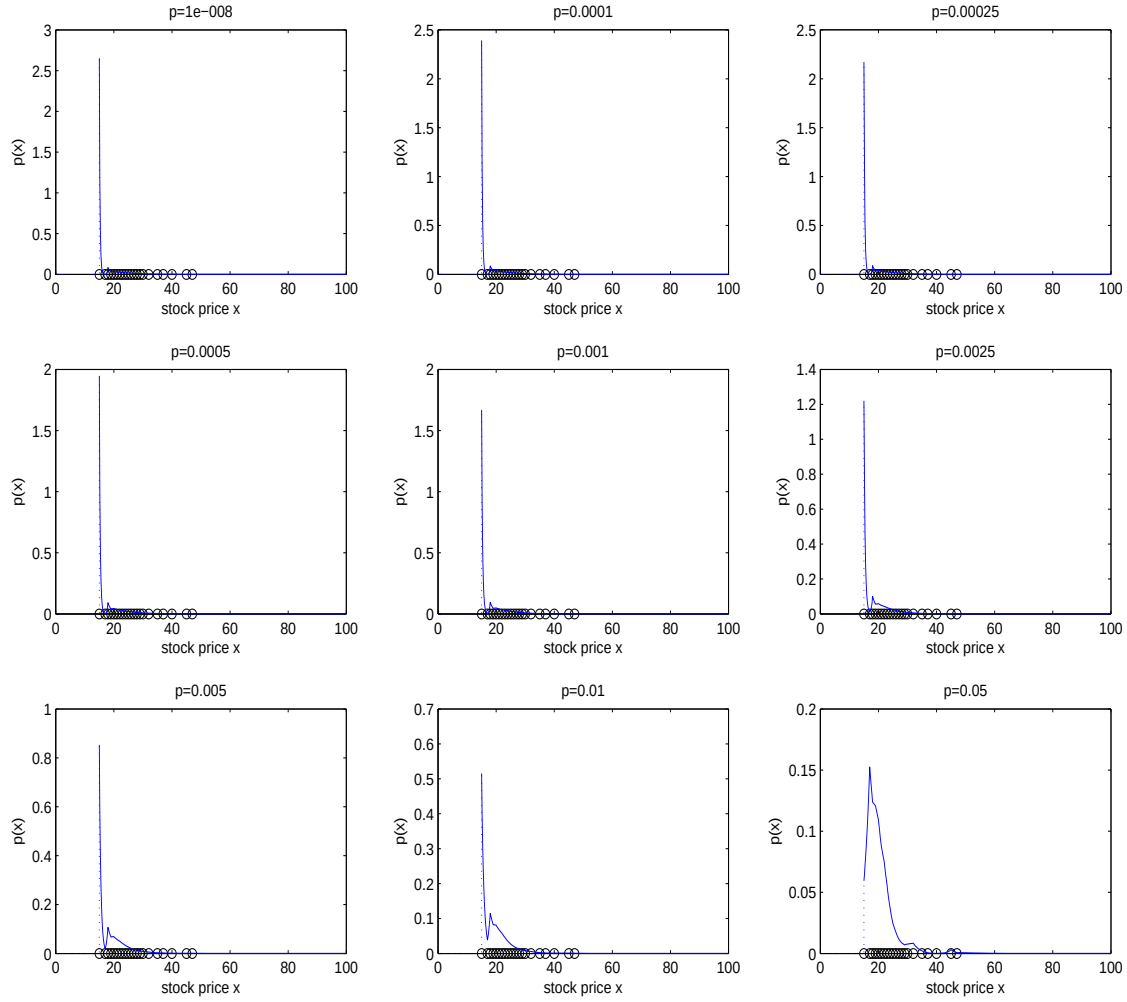

Figure 5.4: The reconstructed cubic spline smoothing density of VIX option index data corresponding to the maximum modulus of $\boldsymbol{\eta}$

Up until now, it can be quite subjective to decide which value of p is more preferred. However,

what is more important in this thesis is that it exhibits the power of maximum entropy principle that risk-neutral densities could be reconstructed nonparametrically. Through the cubic spline smoothing, one can effectively construct a range of densities from various p. I believe by revising this method and including an optimal selection of p would be a new future research direction worth looking at.

Chapter 6

Analysis on Conditioning of the Hessian

For simplicity, assume that $[0, K]$ is a bounded interval (there is no real problem extending to $[0, \infty)$ - one simply takes more care with the allowed λ). Assume $0 \leq k_1 < k_2 < \ldots < k_m \leq K$ are given, and let $\{h_i\}_{i=1}^{m+1}$ be the usual "hat functions" (these are B_1-splines, except h_1 interpolates $\{(0,1),(k_1,1),(k_2,0)\}$ and h_{m+1} interpolates $\{(k_m,0),(K,1)\}$. Note in particular that

$$\sum_{i=1}^{m+1} h_i(x) = \mathbf{1}_{[0,K]}(x). \tag{6.1}$$

The functional to maximise is defined on $\mathbb{R}^{m+1}$ as

$$Q(\lambda) = \eta^T \lambda - \int_0^K e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1}\, dx. \tag{6.2}$$

The optimality conditions for Q are

$$0 = \nabla_\lambda Q(\lambda) = \frac{\partial Q}{\partial \lambda_i} = \eta_i - \int_0^K h_i(x)\; e^{\sum_{k=1}^{m+1} \lambda_i h_i(x) - 1}\, dx \tag{6.3}$$

The Hessian of Q is the symmetric $(m+1) \times (m+1)$ matrix with entries

$$H_{ij} = -\int_0^K h_i(x)\; h_j(x)\; e^{\sum_{k=1}^{m+1} \lambda_i h_i(x) - 1}\, dx$$

For each $\lambda \in \mathbb{R}^{m+1}$ put

$$p_\lambda(x) = e^{\sum_{k=1}^{m+1} \lambda_i h_i(x) - 1}$$

For each $\lambda \in \mathbb{R}^{m+1}$ one can define

$$\langle \phi, \psi \rangle_\lambda := \int_0^K \phi(x)\ \psi(x)\ p_\lambda(x)\, dx.$$

Because $p_\lambda > 0$ on $[0, K]$, this defines a *real inner product* on the continuous functions on $[0, K]$. In particular, $\langle \phi, \phi \rangle_\lambda > 0$ unless $\phi = 0$. Now, for each $a \in \mathbb{R}^{m+1}$, one can define

$$\phi_a(x) = \sum_{i=1}^{m+1} a_i h_i(x)$$

Then

$$a^T H a = \sum_{i,j=1}^{m+1} a_i H_{ij} a_j = -\sum_{i,j} \int_0^K a_i a_j h_i(x) h_j(x)\ e^{\sum_{k=1}^{m+1} \lambda_i h_i(x) - 1}\, dx = -\langle \phi_a, \phi_a \rangle_\lambda.$$

Since $-\langle \phi_a, \phi_a \rangle_\lambda < 0$ unless $\phi_a = 0$ and from (6.1) we see that the $\{h_1, h_2, \ldots, h_{m+1}\}$ are linear independent. Thus this proves that the Hessian H is **negative definite**. And hence, H^{-1} exists.

Since H is a real symmetric and definite negative matrix, it can be written as

$$H = -A^2$$

where A is symmetric.

6.1 Conditioning of H at the Optimum

Let us now focus on the components of the Hessian associated with $i = 1, \ldots, m+1$. Suppose that λ^* solves (6.3) and put $p^*(x) = p_{\lambda^*}(x)$. Note also that since each $0 \leq h_i(x) \leq 1$ we have $0 \leq h_i^2(x) \leq h_i(x)$. Then at λ^*

$$e_i^T H e_i = H_{ii} = -\int_0^K h_i^2(x) p^*(x)\, dx \geq -\int_0^K h_i(x) p^*(x)\, dx = -\eta_i$$

where e_i is the basis vector with 1 in the ith component, and zeros elsewhere.

Put another way,

$$|e_i^T H e_i| \leq |\eta_i| \|e_i\|^2.$$

Since $H = -A^2$, write $e_i = A^{-1}y$ and observe that

$$|y^T H^{-1} y| = |e_i^T e_i| = \|e_i\|^2 \geq \frac{1}{\eta_i}|e_i^T H e_i| = \frac{1}{\eta_i}\|y\|^2.$$

Since H^{-1} is symmetric,

$$\|H^{-1}\| = \sup_z \frac{|z^T H^{-1} z|}{\|z\|^2} > \frac{1}{\min_i \eta_i}. \tag{6.4}$$

The condition number, $\kappa(H)$ of the Hessian matrix H is given by

$$\kappa(H) = \|H^{-1}\| \cdot \|H\|.$$

If the condition number of a matrix turns out to be very large (for example a order above 10^4), we can conclude the matrix is **ill-conditioned**. Ill-conditioned matrix could cause optimisation problem fail to converge to a solution λ^*. From (6.4), we can see that very small η could easily blow up $\|H^{-1}\|$ which leads to bad conditioning of the Hessian.

6.2 Conditioning of H Near the Optimum

If we define a new function $\eta_i(\lambda)$ for any λ as follows

$$\eta_i(\lambda) := \int_0^K h_i(x)\ p_\lambda(x)\, dx.$$

Then we can apply the argument in (6.4) as $\|H^{-1}\| > \frac{1}{\min_i \eta_i(\lambda)}$. The condition number will thus blow up along any sequence of λ which is attempting to converge to a solution of (6.3) where $\eta_i \leq 0$.

From these two sections, it is clear that the optimisation may fail to converge when $\eta_i \leq 0$ or when η gets very close to 0. In case we know in advance that $\eta_i = 0$, the domain restriction excises appropriate intervals. The effect on integrals is to eliminate any contribution from the support of h_i. This certainly sets the ith row and column of H to zero, causing an additional singular direction to appear in H. We will then have a singular H. By completely removing the ith rows and columns where $\eta_i = 0$, we can eventually get a more robust system for the optimisation.

6.3 Regularisation

By imposing a Tikhonov regularisation, we are essentially adding an extra penalty term to (6.2) such that

$$Q_\alpha(\lambda) = Q + \alpha\|\lambda\|^2.$$

Thus, the regularised optimality condition becomes:

$$0 = \frac{\partial Q}{\partial \lambda_i} + 2\alpha\lambda_i$$

Then the Hessian matrix will just be

$$H_\alpha = H + 2\alpha I$$

This will shift all the eigenvalues of H by 2α. Since the eigenvalues of H are all negative, the eigenvalues of H_α must all be less than 2α. Then $\|H_\alpha^{-1}\| \leq \frac{1}{2\alpha}$ and $\|H_\alpha\| = \|H\| + 2\alpha$.

Together, the condition number of H_α is just

$$\begin{aligned}\kappa(H_\alpha) &= \|H_\alpha^{-1}\| \cdot \|H_\alpha\| \\ &\leq \frac{1}{2\alpha} \cdot (\|H\| + 2\alpha) \\ &= 1 + \frac{\|H\|}{2\alpha}\end{aligned}$$

Thus, the regularised optimisation problem will become better conditioned when the value α is selected where the magnitude of $\frac{\|H\|}{2\alpha}$ is controlled.

Chapter 7

Conclusion

Option pricing is an important issue as products need to be priced fairly. Failing to do so would cause arbitrage opportunities to emerge or even false belief of the price in the market. Black-Scholes pricing model (see [5] and [23]) is a commonly used model to price an option by assuming the underlying stock to follow a geometric Brownian motion with constant drift and volatility where the assumptions turned out to be wrong. The non-constant volatility suggests that the Black-Scholes price was mispriced and need to be calibrated either by replacing the volatility measure with either implied volatility or stochastic volatility. An alternative way to look at this is instead of assuming a movement in the stock price, it is assumed to be held fixed. By only taking the market option prices data, one wishes to reconstruct a risk-neutral density from them nonparametrically and take the expectation payoff to obtain the fair prices. One of the most popular ways to pick the risk-neutral density is via the maximum entropy principle.

The maximum entropy principle makes use of a data set of option prices and selects a risk-neutral density where its entropy is maximised. Following the original Buchen & Kelly in [10] their formulation of the maximum entropy problem through the Lagrangian may not guarantee the existence of a solution when the problem itself is infinite-dimensional. Borwein et al. [7] suggests the solution exists in the dual problem under convex duality, by making use of the Lagrange multipliers as new variables for the optimisation problem. Bose & Murray [8] then transformed the basis function of the option prices to hat functions which has finite support instead of the payoff functions which do not. Having a new hat basis function not only gives us the idea of η which allows us to explore non-strictly convex (Type B arbitrage) and non-convex (Type A arbitrage) data sets for an early sign of potential non-convergence in the optimisation problem. During this process, the existence of a maximum entropy solution comes down to checking whether the data set is strictly convex and noiseless. In the real market, the data sets are often non-strictly convex and noisy.

This thesis looks for an effective strategy to obtain maximum entropy solution from the real data set which are non-strictly convex and noisy. By applying the arbitrage-free principle, if one cannot

guarantee its strong form to hold (absence of Type A arbitrage), then its weak form must hold (absence of Type B arbitrage). Therefore, by data linearising, a method which I proposed in this thesis, it is possible to smooth those non-convex data to a non-strictly convex (that is with $\eta = 0$) and to be removed from the domain of the strike prices and underlying prices.

In order to deal with noisy and non-convex data, various strategies including the Tikhonov regularisation, polyhedral set projection method, convex hull method and cubic spline smoothing method have been tested with 1000 replications of simulated experiment performed with programs written by myself in MATLAB. In each experiment, a log-normal option prices with arbitrary parameters were first chosen and added by randomly generated white noises. By exploring the four strategies, it is tested by minimising the difference between the original noiseless density and the risk-neutral density recovered after applying the respective strategy on each simulated noisy data set. The difference measure here is chosen to be the mean integrated squared error. As a result, it is shown that out of 1000 replications, the Tikhonov regularisation, projection method and the convex hull method had indicated that a flat distribution (approaching uniform distribution) is preferred. This suggests that the data set provided do not have much significance in the density reconstruction. Fortunately, with the cubic spline smoothing method, it is found that the density reconstructed could have very similar features to the original noiseless density with the mean integrated squared error to be around 0.002. This method is also very flexible in a way that choosing the mixing parameter p carefully, one can achieve a certain level of smoothness - together with the data linearising method, one can create a convex data satisfying the constraint qualification for the optimisation. The smoothness is also good in a way that it also smooths out the level of noise. Hence, the cubic spline smoothing method is an effective method I have found in this project.

Moreover, I have also discovered how small η could result in an ill-conditioned Hessian matrix which could easily lead the solution failing to converge. It is then recommended to set those η to zero and remove the corresponding price intervals and rows and columns in the Hessian matrix. Alternatively, an Tikhonov regularisation could immediately solve this problem as the condition number will be bounded by the factor of $\frac{1}{\alpha}$, hence, if the chosen α is large enough, one can form a well-conditioned regularised optimisation problem to solve.

Finally, the cubic spline smoothing method is applied to the real data, VIX, a volatility index of the Standard & Poor 500 stocks. The data is then processed in a way that the modulus of the η of the sub-data was measured. Three pieces of sub-data were pulled out to represent: most noisy data, median noisy data and minimum noisy data where the noisy levels of those three data sets were measured by the modulus of η. As expected more noisy data required more smoothing and hence the mixing parameter p tends to be smaller. A range of densities were reconstructed for these three data sets with various values of p. So far, the p chosen has been points ranging between 0 and 1 with more points considered between 0 and 0.1, it will be interesting if one could extend the material in this thesis to find an automatic way of choosing p such that the density reconstructed is optimal.

Appendix A

Appendix

A.1 CDF

The **risk-neutral density** (RND), $p(x)$, has the form of:

$$p(x) = \int_{I_0} e^{\sum_{i=1}^{m+1} \lambda_i^* h_i(x) - 1} \, dx$$

and with calculus, one can derive the **cumulative distribution function** (CDF) for the price of the underlying as follow (λ is used in place of λ^*)

For $t \in [k_{j-1}, k_j)$, the CDF $F(t)$ is given as

$$F(t) = \sum_{i=1}^{j-2} (k_{i+1} - k_i) e^{\lambda_i - 1} \frac{e^{\lambda_{i+1} - \lambda_i - 1}}{\lambda_{i+1} - \lambda_i} + (k_j - k_{j-1}) e^{\lambda_{j-1} - 1} \frac{e^{\frac{\lambda_j - \lambda_{j-1}}{k_j - k_{j-1}}(t - k_{j-1})} - 1}{\lambda_j - \lambda_{j-1}}$$

proof:

For $t \in [k_{j-1}, k_j)$:

$$
\begin{aligned}
F(t) &= \int_{-\infty}^{t} p(x)\,dx \\
&= \int_{0}^{t} e^{\sum_{i=1}^{m+1}\lambda_i h_i(x)-1}\,dx \\
&= \int_{0}^{k_{j-1}} e^{\sum_{i=1}^{m+1}\lambda_i h_i(x)-1}\,dx + \int_{k_{j-1}}^{t} e^{\sum_{i=1}^{m+1}\lambda_i h_i(x)-1}\,dx \\
&= \left(\sum_{i=1}^{j-2}\int_{k_i}^{k_{i+1}} e^{\lambda_i h_i(x)+\lambda_{i+1}h_{i+1}(x)-1}\,dx\right) + \int_{k_{j-1}}^{t} e^{\lambda_{j-1}h_{j-1}(x)+\lambda_j h_j(x)-1}\,dx \\
&= \sum_{i=1}^{j-2}\left(\int_{k_i}^{k_{i+1}} e^{\lambda_i\left(1-\frac{x-k_i}{k_{i+1}-k_i}\right)+\lambda_{i+1}\left(\frac{x-k_i}{k_{i+1}-k_i}\right)-1}\,dx\right) + \int_{k_{j-1}}^{t} e^{\lambda_{j-1}\left(1-\frac{x-k_{j-1}}{k_j-k_{j-1}}\right)+\lambda_j\left(\frac{x-kj-1}{k_j-k_{j-1}}\right)-1}\,dx \\
&= \sum_{i=1}^{j-2}(k_{i+1}-k_i)e^{\lambda_i-1}\frac{e^{\lambda_{i+1}-\lambda_i-1}}{\lambda_{i+1}-\lambda_i} + (k_j-k_{j-1})e^{\lambda_{j-1}-1}\frac{e^{\frac{\lambda_j-\lambda_{j-1}}{k_j-k_{j-1}}(t-k_{j-1})}-1}{\lambda_j-\lambda_{j-1}}
\end{aligned}
$$

□

Note:

If for any i, $\lambda_i = \lambda_{i+1}$, then $\frac{e^{\lambda_{i+1}-\lambda_i-1}}{\lambda_{i+1}-\lambda_i}$ is just 1 by taking the limit of $\lambda_i \to \lambda_{i+1}$. Thus, the term $(k_{i+1}-k_i)e^{\lambda_i-1}\frac{e^{\lambda_{i+1}-\lambda_i-1}}{\lambda_{i+1}-\lambda_i}$ can be simplified to $(k_{i+1}-k_i)e^{\lambda_i-1}$.

If $\lambda_{j-1} = \lambda_j$, then $\frac{e^{\frac{\lambda_j-\lambda_{j-1}}{k_j-k_{j-1}}(t-k_{j-1})}-1}{\lambda_j-\lambda_{j-1}}$ is replaced by $\frac{t-k_{j-1}}{k_j-k_{j-1}}$.

Now if one chooses $t = K$, the entire domain is being calculated now. So $t \in (k_m, K)$ where j takes the value $m+1$ (since $k_{m+1} := K$ and one arrives at the following relation

$$\int_{0}^{K} e^{\sum_{i=1}^{m+1}\lambda_i h_i(x)-1}\,dx = \sum_{i=1}^{m}(k_{i+1}-k_i)e^{\lambda_i-1}\frac{e^{\lambda_{i+1}-\lambda_i-1}}{\lambda_{i+1}-\lambda_i} \tag{A.1}$$

A.2 Q, $\nabla_\lambda Q(\lambda)$, and $\nabla^2_\lambda Q(\lambda)$

MES is the solution to the following problem:

$$\max Q(\lambda) = \langle \lambda, \eta \rangle - \int_{0}^{K} e^{\sum_{i=1}^{m+1}\lambda_i h_i(x)-1}\,dx$$

achieved by setting the $\partial Q = 0$, which is:

$$\nabla_\lambda Q(\lambda) = \frac{\partial Q}{\partial \lambda_i} = \eta_i - \int_0^K h_i(x)\ e^{\sum_{k=1}^{m+1} \lambda_i h_i(x) - 1}\, dx$$

We will begin with Q. By applying the relation in A.1, we can get

$$\begin{aligned} Q &= \langle \lambda, \eta \rangle - \int_0^K e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1}\, dx \\ &= \langle \lambda, \eta \rangle - \sum_{i=1}^{m} (k_{i+1} - k_i) e^{\lambda_i - 1} \frac{e^{\lambda_{i+1} - \lambda_i} - 1}{\lambda_{i+1} - \lambda_i} \end{aligned}$$

If $\lambda_i = \lambda_{i+1}$, the term $(k_{i+1} - k_i)e^{\lambda_i - 1}\frac{e^{\lambda_{i+1}-\lambda_i}-1}{\lambda_{i+1}-\lambda_i}$ will be replaced by $(k_{i+1} - k_i)e^{\lambda_i - 1}$. We can see this as follows:

By fixing λ_{i+1}

$$\begin{aligned} &\lim_{\lambda_i \to \lambda_{i+1}} \frac{e^{\lambda_{i+1} - \lambda_i} - 1}{\lambda_{i+1} - \lambda_i} \\ &= \lim_{\lambda_i \to \lambda_{i+1}} \frac{-e^{\lambda_{i+1} - \lambda_i}}{-1} \qquad \text{By L'Hôpital's Rule} \\ &= 1 \end{aligned}$$

Now we can easily derive $\nabla_\lambda Q(\lambda)$ directly from Q with calculus. We will start with the case when $\lambda_i \neq \lambda_{i+1}$:

$$\begin{aligned} \frac{\partial Q}{\partial \lambda_i} &= \frac{\partial}{\partial \lambda_i} \left(\langle \lambda, \eta \rangle - \sum_{j=1}^{m} (k_{j+1} - k_j) e^{\lambda_j - 1} \frac{e^{\lambda_{j+1} - \lambda_j} - 1}{\lambda_{j+1} - \lambda_j} \right) \\ &= \frac{\partial}{\partial \lambda_i} \left(\langle \lambda, \eta \rangle - \left((k_i - k_{i-1}) e^{\lambda_{i-1} - 1} \frac{e^{\lambda_i - \lambda_{i-1}} - 1}{\lambda_i - \lambda_{i-1}} + (k_{i+1} - k_i) e^{\lambda_i - 1} \frac{e^{\lambda_{i+1} - \lambda_i} - 1}{\lambda_{i+1} - \lambda_i} \right) \right) \\ &= \eta_i - \left\{ (k_i - k_{i-1}) e^{\lambda_{i-1} - 1} \frac{(\lambda_i - \lambda_{i-1}) e^{\lambda_i - \lambda_{i-1}} - \left(e^{\lambda_i - \lambda_{i-1}} - 1 \right)}{(\lambda_i - \lambda_{i-1})^2} \right. \\ &\qquad \left. + (k_{i+1} - k_i) e^{\lambda_i - 1} \left[\frac{e^{\lambda_{i+1} - \lambda_i} - 1}{\lambda_{i+1} - \lambda_i} + \frac{-(\lambda_{i+1} - \lambda_i) e^{\lambda_{i+1} - \lambda_i} + \left(e^{\lambda_{i+1} - \lambda_i} - 1 \right)}{(\lambda_{i+1} - \lambda_i)^2} \right] \right\} \\ &= \eta_i - (k_i - k_{i-1}) e^{\lambda_{i-1} - 1} \frac{(\lambda_i - \lambda_{i-1}) e^{\lambda_i - \lambda_{i-1}} - e^{\lambda_i - \lambda_{i-1}} + 1}{(\lambda_i - \lambda_{i-1})^2} \\ &\qquad - (k_{i+1} - k_i) e^{\lambda_i - 1} \frac{-(\lambda_{i+1} - \lambda_i) + e^{\lambda_{i+1} - \lambda_i} - 1}{(\lambda_{i+1} - \lambda_i)^2} \end{aligned} \tag{A.2}$$

For the cases when $\lambda_{i-1} = \lambda_i$ and $\lambda_i = \lambda_{i+1}$, the following terms will replace the respective terms in (A.2)

- When $\lambda_{i-1} = \lambda_i$, we fix λ_{i-1} again, then

$$\begin{aligned}
&\lim_{\lambda_i \to \lambda_{i-1}} \frac{(\lambda_i - \lambda_{i-1})e^{\lambda_i - \lambda_{i-1}} - e^{\lambda_i - \lambda_{i-1}} + 1}{(\lambda_i - \lambda_{i-1})^2} \\
&= \lim_{\lambda_i \to \lambda_{i-1}} \frac{(\lambda_i - \lambda_{i-1})e^{\lambda_i - \lambda_{i-1}}}{2(\lambda_i - \lambda_{i-1})} && \text{(L'Hôpital's Rule)} \\
&= \lim_{\lambda_i \to \lambda_{i-1}} \frac{(\lambda_i - \lambda_{i-1})e^{\lambda_i - \lambda_{i-1}} + e^{\lambda_i - \lambda_{i-1}}}{2} && \text{(L'Hôpital's Rule)} \\
&= \frac{1}{2}
\end{aligned}$$

- When $\lambda_i = \lambda_{i+1}$, then by fixing λ_{i+1} we have

$$\begin{aligned}
&\lim_{\lambda_i \to \lambda_{i+1}} \frac{-(\lambda_{i+1} - \lambda_i) + e^{\lambda_{i+1} - \lambda_i} - 1}{(\lambda_{i+1} - \lambda_i)^2} \\
&= \lim_{\lambda_i \to \lambda_{i+1}} \frac{1 - e^{\lambda_{i+1} - \lambda_i}}{-2(\lambda_{i+1} - \lambda_i)} && \text{(L'Hôpital's Rule)} \\
&= \lim_{\lambda_i \to \lambda_{i+1}} \frac{e^{\lambda_{i+1} - \lambda_i}}{2} && \text{(L'Hôpital's Rule)} \\
&= \frac{1}{2}
\end{aligned}$$

The Hessian matrix, H, can also be derived by differentiating $\nabla_\lambda Q(\lambda)$, and note that this matrix is a tridiagonal sparse matrix where the only non-zero terms are $\frac{\partial^2 Q}{\partial \lambda_i^2}$ and the $\frac{\partial^2 Q}{\partial \lambda_j \partial \lambda_i}$ when $j = i+1$ or $i-1$.

Again we will split these into the following cases:

$$\nabla^2_\lambda Q(\lambda) = \nabla(\nabla_\lambda Q(\lambda)) = \frac{\partial^2 Q}{\partial \lambda_j \partial \lambda_i} = -\int_0^K h_i(x)\ h_j(x)\ e^{\sum_{i=1}^{m+1} \lambda_i h_i(x) - 1}\, dx$$

For $2 \leq i \leq m$

$$\begin{aligned}
\frac{\partial^2 Q}{\partial \lambda_i^2} &= \frac{\partial}{\partial \lambda_i}\Bigg(\eta_i - (k_i - k_{i-1})e^{\lambda_{i-1}-1}\frac{(\lambda_i - \lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} - e^{\lambda_i-\lambda_{i-1}} + 1}{(\lambda_i - \lambda_{i-1})^2} \\
&\qquad - (k_{i+1} - k_i)e^{\lambda_i - 1}\frac{-(\lambda_{i+1} - \lambda_i) + e^{\lambda_{i+1}-\lambda_i} - 1}{(\lambda_{i+1} - \lambda_i)^2}\Bigg) \\
&= -\Bigg\{(k_i - k_{i-1})e^{\lambda_{i-1}-1}\frac{(\lambda_i - \lambda_{i-1})^3 e^{\lambda_i-\lambda_{i-1}} - 2(\lambda_i - \lambda_{i-1})\left[(\lambda_i - \lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} - e^{\lambda_i-\lambda_{i-1}} + 1\right]}{(\lambda_i - \lambda_{i-1})^4} \\
&\quad + (k_{i+1} - k_i)e^{\lambda_i - 1}\left[\frac{-(\lambda_{i+1} - \lambda_i) + e^{\lambda_{i+1}-\lambda_i} - 1}{(\lambda_{i+1} - \lambda_i)^2} + \frac{\left(1 - e^{\lambda_{i+1}-\lambda_i}\right) + 2\left(-(\lambda_{i+1} - \lambda_i) + e^{\lambda_{i+1}-\lambda_i} - 1\right)}{(\lambda_{i+1} - \lambda_i)^3}\right]\Bigg\} \\
&= -\Bigg[(k_i - k_{i-1})e^{\lambda_{i-1}-1}\frac{(\lambda_i - \lambda_{i-1})^2 e^{\lambda_i-\lambda_{i-1}} - 2(\lambda_i - \lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} + 2e^{\lambda_i-\lambda_{i-1}} - 2}{(\lambda_i - \lambda_{i-1})^3} \\
&\qquad + (k_{i+1} - k_i)e^{\lambda_i - 1}\frac{-(\lambda_{i+1} - \lambda_i)^2 - 2(\lambda_{i+1} - \lambda_i) + 2e^{\lambda_{i+1}-\lambda_i} - 2}{(\lambda_{i+1} - \lambda_i)^3}\Bigg] \qquad \text{(A.3)}
\end{aligned}$$

We will once again consider the cases when $\lambda_{i-1} = \lambda_i$ and $\lambda_i = \lambda_{i+1}$.

- When $\lambda_{i-1} = \lambda_i$, we fix λ_{i-1} again, then

$$\begin{aligned}
&\lim_{\lambda_i \to \lambda_{i-1}} \frac{(\lambda_i - \lambda_{i-1})^2 e^{\lambda_i-\lambda_{i-1}} - 2(\lambda_i - \lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} + 2e^{\lambda_i-\lambda_{i-1}} - 2}{(\lambda_i - \lambda_{i-1})^3} \\
&= \lim_{\lambda_i \to \lambda_{i-1}} \frac{(\lambda_i - \lambda_{i-1})^2}{3(\lambda_i - \lambda_{i-1})^2} \qquad \text{(L'Hôpital's Rule)} \\
&= \lim_{\lambda_i \to \lambda_{i-1}} \frac{2(\lambda_i - \lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} + (\lambda_i - \lambda_{i-1})^2 e^{\lambda_i-\lambda_{i-1}}}{6(\lambda_i - \lambda_{i-1})} \qquad \text{(L'Hôpital's Rule)} \\
&= \lim_{\lambda_i \to \lambda_{i-1}} \frac{2(\lambda_i - \lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} + 2e^{\lambda_i-\lambda_{i-1}} + 2(\lambda_i - \lambda_{i-1})^2 e^{\lambda_i-\lambda_{i-1}} + 2(\lambda_i - \lambda_{i-1})e^{\lambda_i-\lambda_{i-1}}}{6} \\
&= \frac{1}{3}
\end{aligned}$$

- When $\lambda_i = \lambda_{i+1}$, then by fixing λ_{i+1} we have

$$\begin{aligned}
&\lim_{\lambda_i \to \lambda_{i+1}} \frac{-(\lambda_{i+1}-\lambda_i)^2 - 2(\lambda_{i+1}-\lambda_i) + 2e^{\lambda_{i+1}-\lambda_i} - 2}{(\lambda_{i+1}-\lambda_i)^3} \\
&= \lim_{\lambda_i \to \lambda_{i+1}} \frac{2(\lambda_{i+1}-\lambda_i) + 2 - 2e^{\lambda_{i+1}-\lambda_i}}{3(\lambda_{i+1}-\lambda_i)^2} && \text{(L'Hôpital's Rule)} \\
&= \lim_{\lambda_i \to \lambda_{i+1}} \frac{-2 + 2e^{\lambda_{i+1}-\lambda_i}}{-6(\lambda_{i+1}-\lambda_i)} && \text{(L'Hôpital's Rule)} \\
&= \lim_{\lambda_i \to \lambda_{i+1}} \frac{-2e^{\lambda_{i+1}-\lambda_i}}{6} && \text{(L'Hôpital's Rule)} \\
&= -\frac{1}{3}
\end{aligned}$$

Then we know for $i = 1$ is just the second half of Equation (A.3), that is

$$\frac{\partial^2 Q}{\partial \lambda_1^2} = (k_2 - k_1)e^{\lambda_1 - 1} \frac{-(\lambda_2-\lambda_1)^2 - 2(\lambda_2-\lambda_1) + 2e^{\lambda_2-\lambda_1} - 2}{(\lambda_2-\lambda_1)^3}$$

If $\lambda_1 = \lambda_2$

$$\frac{\partial^2 Q}{\partial \lambda_1^2} = -\frac{1}{3}(k_2 - k_1)e^{\lambda_1 - 1}$$

Also for $i = m + 1$ as the first half of Equation (A.3), we have

$$\frac{\partial^2 Q}{\partial \lambda_{m+1}^2} = (k_{m+1} - k_m)e^{\lambda_m - 1} \frac{(\lambda_{m+1}-\lambda_m)^2 e^{\lambda_{m+1}-\lambda_m} - 2(\lambda_{m+1}-\lambda_m)e^{\lambda_{m+1}-\lambda_m} + 2e^{\lambda_{m+1}-\lambda_m} - 2}{(\lambda_{m+1}-\lambda_m)^3}$$

If $\lambda_m = \lambda_{m+1}$

$$\frac{\partial^2 Q}{\partial \lambda_{m+1}^2} = \frac{1}{3}(k_{m+1} - k_m)e^{\lambda_m - 1}$$

From Equation (A.2), we can see that $\frac{\partial^2 Q}{\partial \lambda_j \partial \lambda_i}$ will be mostly zeros unless $j = i + 1$ or $j = i - 1$. Let's look at each of the case.

- For $1 \le i \le m$

$$\begin{aligned}
\frac{\partial^2 Q}{\partial \lambda_{i+1} \partial \lambda_i} &= -\left[(k_{i+1} - k_i)e^{\lambda_i - 1} \frac{(\lambda_{i+1}-\lambda_i)\left(-1 + e^{\lambda_{i+1}-\lambda_i}\right) - 2\left(-(\lambda_{i+1}-\lambda_i) + e^{\lambda_{i+1}-\lambda_i} - 1\right)}{(\lambda_{i+1}-\lambda_i)^3} \right] \\
&= -(k_{i+1} - k_i)e^{\lambda_i - 1} \frac{(\lambda_{i+1}-\lambda_i)e^{\lambda_{i+1}-\lambda_i} - 2e^{\lambda_{i+1}-\lambda_i} + (\lambda_{i+1}-\lambda_i) + 2}{(\lambda_{i+1}-\lambda_i)^3}
\end{aligned}$$

When $\lambda_i = \lambda_{i+1}$

$$\begin{aligned}
&\lim_{\lambda_{i+1}\to\lambda_i} \frac{(\lambda_{i+1}-\lambda_i)e^{\lambda_{i+1}-\lambda_i} - 2e^{\lambda_{i+1}-\lambda_i} + (\lambda_{i+1}-\lambda_i) + 2}{(\lambda_{i+1}-\lambda_i)^3} \\
&= \lim_{\lambda_{i+1}\to\lambda_i} \frac{(\lambda_{i+1}-\lambda_i)e^{\lambda_{i+1}-\lambda_i} - e^{\lambda_{i+1}-\lambda_i} + 1}{3(\lambda_{i+1}-\lambda_i)^2} && \text{(L'Hôpital's Rule)} \\
&= \lim_{\lambda_{i+1}\to\lambda_i} \frac{(\lambda_{i+1}-\lambda_i)e^{\lambda_{i+1}-\lambda_i}}{6(\lambda_{i+1}-\lambda_i)} && \text{(L'Hôpital's Rule)} \\
&= \frac{1}{6}
\end{aligned}$$

then $\dfrac{\partial^2 Q}{\partial\lambda_{i+1}\partial\lambda_i} = -\dfrac{1}{6}(k_{i+1}-k_i)e^{\lambda_i - 1}$

- For $2 \le i \le m+1$

$$\begin{aligned}
\frac{\partial^2 Q}{\partial\lambda_{i-1}\partial\lambda_i} &= -\Bigg\{(k_i - k_{i-1})e^{\lambda_{i-1}-1}\Bigg[\frac{(\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} - e^{\lambda_i-\lambda_{i-1}} + 1}{(\lambda_i-\lambda_{i-1})^2} \\
&\qquad + \frac{-(\lambda_i-\lambda_{i-1})^2 e^{\lambda_i-\lambda_{i-1}}}{(\lambda_i-\lambda_{i-1})^3} + \frac{2((\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} - e^{\lambda_i-\lambda_{i-1}} + 1)}{(\lambda_i-\lambda_{i-1})^3}\Bigg]\Bigg\} \\
&= -(k_i-k_{i-1})e^{\lambda_{i-1}-1}\frac{(\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} - 2e^{\lambda_i-\lambda_{i-1}} + (\lambda_i-\lambda_{i-1}) + 2}{(\lambda_i-\lambda_{i-1})^3}
\end{aligned}$$

When $\lambda_{i-1} = \lambda_i$

$$\begin{aligned}
&\lim_{\lambda_i\to\lambda_{i-1}} \frac{(\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} - 2e^{\lambda_i-\lambda_{i-1}} + (\lambda_i-\lambda_{i-1}) + 2}{(\lambda_i-\lambda_{i-1})^3} \\
&= \lim_{\lambda_i\to\lambda_{i-1}} \frac{(\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}} - e^{\lambda_i-\lambda_{i-1}} + 1}{3(\lambda_i-\lambda_{i-1})^2} && \text{(L'Hôpital's Rule)} \\
&= \lim_{\lambda_i\to\lambda_{i-1}} \frac{(\lambda_i-\lambda_{i-1})e^{\lambda_i-\lambda_{i-1}}}{6(\lambda_i-\lambda_{i-1})} && \text{(L'Hôpital's Rule)} \\
&= \frac{1}{6}
\end{aligned}$$

then $\dfrac{\partial^2 Q}{\partial\lambda_{i-1}\partial\lambda_i} = -\dfrac{1}{6}(k_i-k_{i-1})e^{\lambda_{i-1}-1}$

From this, we see that the non-linear integral computation has now been turned into sums of linear terms. This has not only made the computation faster and also permit rooms for computing larger value of m more efficiently.

A.3 Bias Correction for Cubic Spline Smoothing Method

As shown by Newman in 1993 [25], suppose there is data set X and Y and

$$log(Y) = \beta_0 + \beta_1 X + \epsilon$$

is the model specification of Y and the fitted value of Y is $e^{\beta_0+\beta_1 X}$, this has bias as

$$Y = e^{\beta_0+\beta_1 X+\epsilon} = e^{\beta_0+\beta_1 X}\ e^{\epsilon}$$

Hence, this leads to the estimation of e^{ϵ}, which can be estimated by

$$e^{\epsilon} = e^{MSE/2}$$

where MSE is the mean squared error (MSE) of the model.

The MSE of the cubic spline smoothing can be estimated using the variance estimator. Before doing so, some terms will need to be defined first. The following is an excerpt from Eubanks's book([13])

Setting $\lambda = \frac{1-p}{p}$, the problem in (4.5) is just

$$\text{find} \quad f = \arg\min \mu_\lambda = \lambda \int (f''(t))^2 dt + \frac{1}{n}\sum_{i=1}^{n}(y_i - f(t_i))^2$$

μ_λ can be represented as (also in [13])

$$\mu_\lambda = y - n\lambda Q(n\lambda Q^T Q + B)^{-1} Q^T y$$

where Q^T is an $(n-2)\times n$ tridiagonal matrix with i-th row

$$(\underbrace{0,\ldots,0}_{i-1}, \frac{1}{t_{i+1}-t_i}, -\frac{1}{t_{i+2}-t_{i-1}} - \frac{1}{t_{i+1}-t_i}, \frac{1}{t_{i+2}-t_{i+1}}, \underbrace{0,\ldots,0}_{n-i-2})$$

and $6B$ is a symmetric $(n-2)\times(n-2)$ tridiagonal matrix having first and last rows $(t_2 - t_1, t_3 - t_2, \underbrace{0,\ldots,0}_{n-4})$ and $(\underbrace{0,\ldots,0}_{n-4}, (t_{n-1} - t_{n-2}, t_n - t_{n-1})$ and with i-th row

$$(\underbrace{0,\ldots,0}_{i-2}, t_{i+1}-t_i, 2(t_{i+2}-t_i), t_{i+2}-t_{i+1}, \underbrace{0,\ldots,0}_{n-i-3})$$

By arranging $\mu_\lambda = S_\lambda\ y$, the matrix S_λ is just

$$S_\lambda = I - n\lambda Q(n\lambda Q^T Q + B)^{-1} Q^T$$

Now it is ready to define the unbiased variance estimator given by

$$\hat{\sigma}^2_\lambda = \frac{y^T (I - S_\lambda)^2 y}{tr(I - S_\lambda)}$$

Then the biased corrected data will be

$$y_i = e^{f(t_i)}\ e^{\frac{\hat{\sigma}^2_\lambda}{2}}$$

A.4 Matlab Code

This section contains all the MATLAB code used in this thesis.

breaklines

```
function main
logn;
for i = 1:3
    real(i);
end
end

% Runs the log-normal experiments
function logn
sigma=0.5;
data = getData(sigma,21);

N = 1000;
alphaMat = [0 1e-16 0.0001 0.001 0.01 0.1 0.25 0.5 1 5];
ise_r = getSampErr(data, sigma, N, 10,'regularisation',alphaMat);
ise_j = getSampErr(data, sigma, N, 10, 'projection', alphaMat);
ise_c = getSampErr(data, sigma, N, 10, 'convex_hull', alphaMat);

pMat=[1e-8 1e-4 2.5e-4 5e-4 1e-3 2.5e-3 5e-3 1e-2 5e-2 7.5e-2];
pMat=[pMat linspace(0.1,1-1e-16,4)];
ise_p = getSampErr(data, sigma, N, 10, 'spline', 0, pMat);

isePlots(ise_r,N, alphaMat);
isePlots(ise_j,N, alphaMat);
isePlots(ise_c,N, alphaMat);
isePlots(ise_p,N, 0, pMat);
end

% Runs the real VIX data set in Chapter 5
function real(dataSetIndex)
load('etaNeg work')
% [dataCell,moduli]=etaNeg;
mvec=moduli(:);
switch dataSetIndex
    case 1
        mvalue = min(mvec);
    case 2
        mvalue = median(mvec);
    case 3
        mvalue = max(mvec);
end
index=moduli==mvalue;

data=dataCell{index};
data(isnan(data(:,2)),:)=[];

K=100;
```

```
data=sortKD(data,K);

k=data(1:end-1,1);
d=data(2:end,2);

k = [0;k];
d = [25.11;d]; % closing price for VIX

data = [[k;K] [1;d]];
eta=getEta(data);

figure;
subplot(2,1,1)
plot(data(:,1),[data(2:end,2) ;0])
hold on;
plot(data(:,1),[data(2:end,2) ;0],'ro')
hold off;
xlabel('strike price, k')
ylabel('option price, d');
title(sprintf('modulus of eta = %g',mvalue));

subplot(2,1,2)
for i = 1:size(data,1)
    plot([data(i,1) data(i,1)],[0 eta(i)])
    hold on
end
plot(data(:,1),zeros(length(data(:,1))),'r:')
plot(data(:,1), eta, 'ro')
hold off
xlabel('strike price, k')
ylabel('eta');

pMat1=[1e-8 1e-4 2.5e-4 5e-4 1e-3 2.5e-3 5e-3 1e-2 5e-2 7.5e-2];
pMat1=[pMat1 linspace(0.1,1-1e-16,10)];

figure
for i = 1:length(pMat1)
    [lamStar,eta,data_curve]=strategy8(data,0,pMat1(i));
    try
        if length(pMat1)>1
            if dataSetIndex == 1
                subplot(4,5,i);
            elseif dataSetIndex == 2
                subplot(3,3,i);
            else
                subplot(3,3,i);
            end
        end
        plotMES(data_curve,lamStar,eta);
        title(sprintf('p=%g',pMat1(i)));
    catch ME
        continue;
    end
end
```

```
end

% Sort the data with K inserted
function sorted=sortKD(unsortData,K)
k=unsortData(:,1);
d=unsortData(:,2);
sorted=[[k;K] [1;d]];
end

% Processing the data with min, mean, median, etc
function [dataCell,moduli]=etaNeg
[data,dates,strikes]=getRealData('VIXB_trade2.csv');
dataCell=cell(35,7);
etaCell=dataCell;
for i=1:7
    dataCell(1,i)={processData(data,dates,strikes,'min',i,[],[])};
    dataCell(2,i)={processData(data,dates,strikes,'mean',i,[],[])};
    dataCell(3,i)={processData(data,dates,strikes,'median',i,[],[])};
    dataCell(4,i)={processData(data,dates,strikes,'weighted mean',i,[],[])};
    dataCell(5,i)={processData(data,dates,strikes,'weighted median',i,[],[])};
    for j=1:2
        dataCell(5+j,i)={processData(data,dates,strikes,'min',i,j,[])};
        dataCell(7+j,i)={processData(data,dates,strikes,'mean',i,j,[])};
        dataCell(9+j,i)={processData(data,dates,strikes,'median',i,j,[])};
        dataCell(11+j,i)={processData(data,dates,strikes,'weighted mean',i,j,[])};
        dataCell(13+j,i)={processData(data,dates,strikes,'weighted median',i,j,[])};
    end
    for k=1:4
        dataCell(15+k,i)={processData(data,dates,strikes,'min',i,[],k)};
        dataCell(19+k,i)={processData(data,dates,strikes,'mean',i,[],k)};
        dataCell(23+k,i)={processData(data,dates,strikes,'median',i,[],k)};
        dataCell(27+k,i)={processData(data,dates,strikes,'weighted mean',i,[],k)};
        dataCell(31+k,i)={processData(data,dates,strikes,'weighted median',i,[],k)};
    end
end

for ii=1:size(dataCell,1)
    for j=1:7
        final=dataCell{ii,j};
        final(any(isnan(final),2),:) = [];
        k=final(:,1);
        d=final(:,2);

        m=size(k,1);
        K=round(max(k)+(k(m)-k(m-1)));

        % Evaluate the matrix B
        B=zeros(m+1,m+1);
        B(1,1:3)=[1,-1/(k(2)-k(1)),1/(k(2)-k(1))];
        for i=2:(m-1)
            b1=1/(k(i)-k(i-1));
            b2=1/(k(i+1)-k(i));
            B(i,i:(i+2))=[b1,-(b1+b2),b2];
        end
```

```
156         B(m,m:(m+1))=[1/(k(m)-k(m-1)),-(1/(k(m)-k(m-1))+1/(K-k(m)))];
157         B(m+1,m+1)=1/(K-k(m));
158         % Evaluate the moment eta
159         eta=B*[1;d];
160         moduli(ii,j)=sum(abs(eta));
161     end
162 end
163 end
164
165 % Processing the data obtained from getRealData.m
166 function finalData=processData(data,dates,strikes,dataType,dateInd,halfInd,quarterInd)
167 format shortG % produce a more readable format
168 if ~isempty(quarterInd)||~isempty(halfInd)
169     quarterLimit1=60*14+30+105; % starting from 14:30
170     quarterLimit2=quarterLimit1+105;
171     quarterLimit3=quarterLimit2+105;
172     tempTime=datestr(data(:,1));
173     totalMin=zeros(size(data,1),1);
174     minInd=totalMin;
175     halfMinInd=totalMin;
176     for j=1:size(data,1)
177         tempHour=str2double(tempTime(j,13:14));
178         tempMin=str2double(tempTime(j,16:17));
179         totalMin(j)=60*tempHour+tempMin;
180         if totalMin(j) <= quarterLimit1
181             minInd(j)=1;
182             halfMinInd(j)=1;
183         elseif totalMin(j) > quarterLimit1 && totalMin(j) <= quarterLimit2
184             minInd(j)=2;
185             halfMinInd(j)=1;
186         elseif totalMin(j) > quarterLimit2 && totalMin(j) <= quarterLimit3
187             minInd(j)=3;
188             halfMinInd(j)=2;
189         else
190             minInd(j)=4;
191             halfMinInd(j)=2;
192         end
193     end
194     data(:,1)=totalMin;
195     data(:,6)=minInd;
196     data(:,7)=halfMinInd;
197 end
198
199 finalPrice=zeros(length(strikes),1);
200 for i=1:length(strikes);
201     temp=data(data(:,3)==strikes(i),:);
202     if any(temp(:,2)==dates(dateInd))
203         if ~isempty(quarterInd) && isempty(halfInd)
204             tempPrice=temp(temp(:,2)==dates(dateInd) & temp(:,6)==quarterInd,4);
205             tempVolume=temp(temp(:,2)==dates(dateInd) & temp(:,6)==quarterInd,5);
206         elseif ~isempty(halfInd) && isempty(quarterInd)
207             tempPrice=temp(temp(:,2)==dates(dateInd) & temp(:,7)==halfInd,4);
208             tempVolume=temp(temp(:,2)==dates(dateInd) & temp(:,7)==halfInd,5);
209         else
```

```
210             tempPrice=temp(temp(:,2)==dates(dateInd),4);
211             tempVolume=temp(temp(:,2)==dates(dateInd),5);
212         end
213     else
214         tempPrice=NaN;
215     end
216     if ~all(isnan(tempPrice))
217         if strcmpi(dataType,'min')
218             finalPrice(i,1)=min(tempPrice);
219         end
220         if strcmpi(dataType,'mean')
221             finalPrice(i,1)=mean(tempPrice);
222         end
223         if strcmpi(dataType,'median')
224             finalPrice(i,1)=median(tempPrice);
225         end
226         if strcmpi(dataType,'weighted mean')
227             finalPrice(i,1)=sum(tempPrice.*tempVolume)/sum(tempVolume);
228         end
229         if strcmpi(dataType,'weighted median')
230             temp1=[];
231             for k=1:size(tempPrice,1)
232                 temp1=[temp1 tempPrice(k)*ones(1,tempVolume(k))];
233             end
234             finalPrice(i,1)=median(temp1);
235         end
236     else
237         finalPrice(i,1)=NaN;
238     end
239 end
240 finalData=[strikes finalPrice];
241 end
242
243 % Read the VIXB csv file
244 function [data,dates,strikes]=getRealData(filename)
245 [RIC,dateMat,time,offset,price,volume]=textread(filename,'%s %s %s %d %f %d',...
246     'delimiter', ',', 'emptyvalue', NaN);
247 RIC=char(RIC);
248 strike=str2num(RIC(:,9:11)); % extract out the three digit num for strikes
249 dateNum=datenum(dateMat);
250 dateTime=strcat(dateMat, {' '}, time);
251 dateTimeNum=datenum(dateTime);
252 data=[dateTimeNum dateNum strike price volume];
253 data(any(isnan(data),2),:) = []; % remove the rows with NaN's
254 dates=unique(dateNum);
255 strikes=unique(strike);
256 end
257
258 % Obtain the log-normal simulated data
259 function data = getData(sigma,m)
260 [~,~,~,K,mu,sd]=getParam(sigma);
261 if sigma==0.5
262     k=linspace(0,80,m);
263 elseif sigma==0.2
```

```
264     if m == 6 || m == 9
265         k=[0,25,linspace(30,60,m-2)];
266     elseif m == 16
267         k=[0,25,linspace(27.5,60,m-2)];
268     else
269         k=[0,25,linspace(28,60,m-2)];
270     end
271 end
272 k=k';
273 d=zeros(length(k),1);
274 for i=1:length(k)
275     partial_ex=exp(mu+sd^2/2)*(normcdf((mu+sd^2-log(k(i)))/sd,0,1)-...
276         normcdf((mu+sd^2-log(K))/sd,0,1));
277     k_integral=k(i)*(normcdf((log(K)-mu)/sd,0,1)-normcdf((log(k(i))-mu)/sd,0,1));
278     d(i)=partial_ex-k_integral;
279     d(i)=d(i)/logncdf(K,mu,sd);
280 end
281 data = [[k;K],[1;d]];
282 end
283
284 % Obtain the parameters needed for the log-normal simulation
285 function [d1,T,r,K,mu,sd,mMat]=getParam(sigma)
286 d1=40;
287 T=0.5;
288 r=0;
289 K=100;
290 mu = log(d1)+(r-sigma^2/2)*T;
291 sd = sigma*sqrt(T);
292 if sigma == 0.5
293     mMat = [5,9,17,21];
294 elseif sigma == 0.2
295     mMat = [6,9,16,20];
296 end
297 end
298
299 % Calculates the eta vector
300 function eta=getEta(data)
301 d=data(:,2);
302 B=getB(data);
303 eta=B*d;
304 end
305
306 % Calculate the matrix B
307 function B=getB(data)
308 m = size(data,1)-1;
309 k = data(1:m,1);
310 K = data(end,1);
311 B=zeros(m+1,m+1);
312 B(1,1:3)=[1,-1/(k(2)-k(1)),1/(k(2)-k(1))];
313 for i=2:(m-1)
314     b1=1/(k(i)-k(i-1));
315     b2=1/(k(i+1)-k(i));
316     B(i,i:(i+2))=[b1,-(b1+b2),b2];
317 end
```

```
318 B(m,m:(m+1))=[1/(k(m)-k(m-1)),-(1/(k(m)-k(m-1))+1/(K-k(m)))];
319 B(m+1,m+1)=1/(K-k(m));
320 end
321
322 % Perform N replications for specified strategies
323 function intErr=getSampErr(data, sigma, N, sd_factor, strategy, alphaMat, pMat)
324 if nargin == 6
325     pMat = 1e-3;
326 end
327 data_n = data;
328 d=data_n(3:end,2);
329 matSize = max(length(alphaMat),length(pMat));
330 intErr = zeros(N,matSize);
331 for j = 1:matSize
332     for i = 1: N
333         try
334             d_n = d+d.*randn(size(d))/sd_factor;
335             data_n(3:end,2)= d_n;
336             switch strategy
337                 case 'regularisation'
338                     [lamStar,eta_f,data_nc]=strategy10(data_n,alphaMat(j));
339                     if lamStar == 0
340                         continue;
341                     else
342                         plotMES(data_nc,lamStar,eta_f,sigma);
343                         intErr(i,j) = getInt_sqr_err(data_nc,lamStar,sigma,eta_f);
344                     end
345                 case 'convex_hull'
346                     [lamStar,eta_f,data_hull]=strategy9(data_n,alphaMat(j));
347                     if lamStar == 0
348                         continue;
349                     else
350                         plotMES(data_hull,lamStar,eta_f,sigma);
351                         intErr(i,j) = getInt_sqr_err(data_hull,lamStar,sigma,eta_f);
352                     end
353                 case 'projection'
354                     [lamStar,eta_f,data_new]=strategy3(data_n,alphaMat(j));
355                     if lamStar == 0
356                         continue;
357                     else
358                         plotMES(data_new,lamStar,eta_f,sigma);
359                         intErr(i,j) = getInt_sqr_err(data_new,lamStar,sigma,eta_f);
360                     end
361                 case 'spline'
362                     if length(alphaMat) == 1
363                         [lamStar,eta_f,data_curve]=strategy8(data_n,alphaMat,pMat(j));
364                         if lamStar == 0
365                             continue;
366                         else
367                             plotMES(data_curve,lamStar,eta_f,sigma);
368                             intErr(i,j) = getInt_sqr_err(data_curve,lamStar,sigma,eta_f);
369                         end
370                     else
371                         fprintf('The alpha should be a scalar. Please check again. \n');
```

```
372                         return;
373                     end
374                 end
375             catch ME
376                 warning('off');
377                 disp(ME);
378             end
379         end
380     end
381     end
382
383     % Data Linearising Method
384     function eta = setNegEta(value,eta,data_curve)
385     k = data_curve(1:end,1);
386     d = data_curve(2:end,2);
387     m = length(k)-1;
388     i = find(eta<0,1);
389     eta(i)=value;
390     if i == 1
391         eta(i+1)=1-value-(d(i+1)-d(i+2))/(k(i+2)-k(i+1));
392     elseif i == m-1
393         eta(i+1)=1-(i-1)*value-(d(i+1)-0)/(k(i+2)-k(i+1));
394     elseif i == m
395         eta(i)=1;
396     else
397         eta(i+1)=1-(i-1)*value-(d(i+1)-d(i+2))/(k(i+2)-k(i+1));
398     end
399     end
400
401     % Tikhonov Regularisation
402     function [lamStar,eta,data]=strategy10(data,alpha)
403     eta = getEta(data);
404     try
405         [lamStar,data] = computeMES(data,eta,alpha);
406     catch ME
407         lamStar = 0;
408         return;
409     end
410     eta(eta==0)=[];
411     end
412
413     % Polyhedral set projection
414     function [lamStar,eta,data_new]=strategy3(data,alpha)
415     d = data(2:end,2);
416     options = optimset('Algorithm','interior-point');
417     d_new=fmincon(@(x) norm(x-d)^2,d,[],[],[],[],[],[],@(x) nonlcon(x,data), options);
418     data_new=data;
419     data_new(2:end,2)=d_new;
420     eta = getEta(data_new);
421     try
422         [lamStar,data_new] = computeMES(data_new,eta,alpha);
423     catch ME
424         lamStar = 0;
425         return;
```

```
426 end
427 end
428
429 % Convex Hull Method
430 function [lamStar,eta,data_curve]=strategy9(data,alpha)
431 d = data(2:end,2);
432 k = data(1:end-1,1);
433 K=data(end,1);
434 dt = DelaunayTri(k,d);
435 kk = convexHull(dt);
436 convhull=[dt.X(kk,1) dt.X(kk,2)];
437 convhull=convhull(1:end-2,:);
438 k_c = convhull(:,1);
439 d_c = convhull(:,2);
440 data_curve = [[k_c;K] [1;d_c]];
441 eta=getEta(data_curve);
442 value = 0;
443 if eta(1)<0
444     while any(eta<0)
445         eta=setNegEta(value,eta,data_curve);
446     end
447 end
448 try
449     [lamStar,data_curve] = computeMES(data_curve,eta,alpha);
450 catch ME
451     lamStar = 0;
452     return;
453 end
454 eta(eta==0)=[];
455 end
456
457 % Cubic Spline Smoothing Method
458 function [lamStar,eta,data_curve]=strategy8(data,alpha,P)
459 d = data(2:end,2);
460 k = data(1:end-1,1);
461 [pp,p1] = csaps(k,[k log(d)]',P);
462 d_curve=ppval(pp,k);
463 d_curve = d_curve(2,:);
464 d_curve = d_curve';
465 bias = biasCorrection(d,k,p1);
466 d_curve=exp(d_curve)*bias;
467 data_curve=data;
468 data_curve(2:end,2)=d_curve;
469 eta=getEta(data_curve);
470 negInd = 0;
471 negInd2 = 1;
472 if eta(1) < 0
473     while any(eta<0) && (negInd2 -1) == negInd;
474         negInd = find(eta<0,1);
475         eta=setNegEta(0,eta,data_curve);
476         negInd2=find(eta<0,1);
477     end
478 end
479 if sum(eta<0) ~= 0
```

```
480     lamStar=0;
481     return;
482 end
483 [lamStar,data_curve] = computeMES(data_curve,eta,alpha);
484 eta(eta==0)=[];
485 end
486
487 % Calculates the bias from log-transformed cubic spline smoother
488 function [bias,var_hat,S] = biasCorrection(y,t,p)
489 y = log(y);
490 if p == 0
491     lambda = Inf;
492 elseif p == 1
493     lambda = 0;
494 else
495     lambda = (1-p)/p;
496 end
497 n = length(t);
498 Q = zeros(n-2,n);
499 for i = 1 : n-2
500     Q(i,i:i+2) = [1/(t(i+1)-t(i)),-1/(t(i+2)-t(i+1))-1/(t(i+1)-t(i)),1/(t(i+2)-t(i+1))];
501 end
502 Q = Q';
503 B = zeros(n-2,n-2);
504 for i = 1 : n-2
505     if i == 1
506         B(1,1:2) = [t(2)-t(1),t(3)-t(2)];
507     elseif i == n-2
508         B(n-2,n-3 : n-2) = [t(n-1)-t(n-2),t(n)-t(n-1)];
509     else
510         B(i,i-1 : i+1) = [t(i+1)-t(i),2*(t(i+2)-t(i)),t(i+2)-t(i+1)];
511     end
512 end
513 B = 1/6*B;
514 invQB = inv(n*lambda*(Q'*Q )+ B);
515 S = (eye(n)-n*lambda*Q*invQB*Q');
516 var_hat = (y'*(eye(n)-S)^2*y)/trace(eye(n)-S);
517 bias = exp(var_hat/2);
518 end
519
520 % Compute the MES with initialisation and setting impossible regions
521 function [lamStar,data] = computeMES(data,eta,alpha)
522 m=length(eta)-1;
523 zero_eta = find(eta==0);
524 lambda = log(linspace(0.001,0.99,m+1));
525 lambda(eta==0)=0;
526 if ~isempty(zero_eta)
527     lambda(zero_eta)=[];
528     eta(zero_eta) = [];
529     d = data(2:end,2);
530     d(zero_eta) = [];
531     d = [1;d];
532     k = data(:,1);
533     k(zero_eta) = [];
```

```
534     data=[k,d];
535 end
536 lamStar=getMES(data,lambda,eta,alpha);
537 end
538
539 % Performs Newton's method to the value function Q
540 function lamStar=getMES(data,lambda,eta,alpha)
541 f=@(lambda)getQ_2(data,lambda,eta,alpha);
542 options = optimset('Hessian','on','GradObj','on','LargeScale','on',...
543     'Display','iter-detailed','DerivativeCheck','off','TolX',1e-12,...
544     'TolFun',1e-12,'MaxIter',1000);
545 [lamStar,~,exitflag]=fminunc(f,lambda,options);
546 notConverged = [0,-1,-3];
547 if any(exitflag == notConverged)
548     lamStar = 0;
549     return;
550 end
551 end
552
553 % Collect the value functions, gradient, and the Hessian matrix for
554 % getMES function
555 function [Q,gradQ,HessQ] = getQ_2(data,lambda,eta,alpha)
556 Q = getQ(data,lambda,eta,alpha);
557 gradQ = getGradQ(data,lambda,eta,alpha);
558 HessQ = getHessQ(data,lambda,eta,alpha);
559 end
560
561 % Calculate the value function Q
562 function Q = getQ(data,lambda,eta,alpha)
563 k = data(:,1);
564 m = length(eta)-1;
565 phi = zeros(m,1);
566 for i = 1 : m
567     if lambda(i) == lambda(i+1)
568         factor = 1;
569     else
570         factor = (exp(lambda(i+1)-lambda(i))-1)/(lambda(i+1)-lambda(i));
571     end
572     phi(i) = (k(i+1)-k(i))*exp(lambda(i)-1)*factor;
573 end
574 phiTotal = sum(phi);
575 Q = lambda*eta - phiTotal - alpha*norm(lambda)^2;
576 Q = -Q;
577 end
578
579 % Calculates the gradient function
580 function gradQ = getGradQ(data,lambda,eta,alpha)
581 k = data(:,1);
582 m = length(eta) - 1;
583 phiStar = zeros(size(eta));
584 if lambda(1) == lambda(2)
585     phiStar(1) = (1/2)*(k(2)-k(1))*exp(lambda(1)-1);
586 else
587     phiStar(1) = (k(2)-k(1))*exp(lambda(1)-1)*(-(lambda(2)-lambda(1))+...
```

```
588         exp(lambda(2)-lambda(1))-1)/(lambda(2)-lambda(1))^2;
589 end
590 for i = 2 : m
591     part1 = (k(i)-k(i-1))*exp(lambda(i-1)-1);
592     part2 = (k(i+1)-k(i))*exp(lambda(i)-1);
593     if lambda(i-1) == lambda(i)
594         factor1 = 1/2;
595     else
596         factor1 = ((lambda(i)-lambda(i-1))*exp(lambda(i)-lambda(i-1))-...
597             exp(lambda(i)-lambda(i-1))+1)/(lambda(i)-lambda(i-1))^2;
598     end
599     if lambda(i) == lambda(i+1)
600         factor2 = 1/2;
601     else
602         factor2 = (-(lambda(i+1)-lambda(i))+exp(lambda(i+1)-lambda(i))-1)...
603             /(lambda(i+1)-lambda(i))^2;
604     end
605     phiStar(i) = part1*factor1 + part2*factor2;
606 end
607 if lambda(m) == lambda(m+1)
608     phiStar(m+1) = (1/2)*(k(m+1)-k(m))*exp(lambda(m)-1);
609 else
610     phiStar(m+1) = (k(m+1)-k(m))*exp(lambda(m)-1)*((lambda(m+1)-...
611         lambda(m))*exp(lambda(m+1)-lambda(m))-exp(lambda(m+1)-lambda(m))+1)...
612         /(lambda(m+1)-lambda(m))^2;
613 end
614 gradQ = eta - phiStar - 2*alpha*lambda';
615 gradQ = -gradQ;
616 end
617
618 % Calculates the Hessian matrix
619 function HessQ = getHessQ(data,lambda,eta,alpha)
620 k = data(:,1);
621 m = length(eta)-1;
622 HessQ = zeros(m+1);
623 if lambda(1) == lambda(2)
624     HessQ(1,1) = -1/3*(k(2)-k(1))*exp(lambda(1)-1);
625 else
626     HessQ(1,1) = -(k(2)-k(1))*exp(lambda(1)-1)*(-(lambda(2)-lambda(1))^2-...
627         2*(lambda(2)-lambda(1))+2*exp(lambda(2)-lambda(1))-2)/(lambda(2)-lambda(1))^3;
628 end
629 for i = 2 : m
630     part1 = (k(i)-k(i-1))*exp(lambda(i-1)-1);
631     part2 = (k(i+1)-k(i))*exp(lambda(i)-1);
632     if lambda(i-1) == lambda(i)
633         factor1 = 1/3;
634     else
635         factor1 = ((lambda(i)-lambda(i-1))^2*exp(lambda(i)-lambda(i-1))-...
636             2*(lambda(i)-lambda(i-1))*exp(lambda(i)-lambda(i-1))+...
637             2*exp(lambda(i)-lambda(i-1))-2)/(lambda(i)-lambda(i-1))^3;
638     end
639     if lambda(i) == lambda(i+1)
640         factor2 = -1/3;
641     else
```

```
642         factor2 = (-(lambda(i+1)-lambda(i))^2-2*(lambda(i+1)-lambda(i))+...
643             2*exp(lambda(i+1)-lambda(i))-2)/(lambda(i+1)-lambda(i))^3;
644     end
645     HessQ(i,i) = -(factor1*part1+factor2*part2);
646 end
647 if lambda(m) == lambda(m+1)
648     HessQ(m+1,m+1) = 1/3*(k(m+1)-k(m))*exp(lambda(m)-1);
649 else
650     HessQ(m+1,m+1) = -(k(m+1)-k(m))*exp(lambda(m)-1)*((lambda(m+1)-...
651         lambda(m))^2*exp(lambda(m+1)-lambda(m))-2*(lambda(m+1)-...
652         lambda(m))*exp(lambda(m+1)-lambda(m))+2*exp(lambda(m+1)-lambda(m))-2)...
653         /(lambda(m+1)-lambda(m))^3;
654 end
655 for i = 1 : m
656     if lambda(i) == lambda(i+1)
657         HessQ(i+1,i) = -1/6*(k(i+1)-k(i))*exp(lambda(i)-1);
658         HessQ(i,i+1) = HessQ(i+1,i);
659     else
660         HessQ(i+1,i) = -(k(i+1)-k(i))*exp(lambda(i)-1)*((lambda(i+1)-...
661             lambda(i))*exp(lambda(i+1)-lambda(i))-2*exp(lambda(i+1)-lambda(i))+...
662             (lambda(i+1)-lambda(i))+2)/(lambda(i+1)-lambda(i))^3;
663         HessQ(i,i+1) = HessQ(i+1,i);
664     end
665 end
666 HessQ = HessQ - 2*alpha*eye(size(HessQ));
667 HessQ = -HessQ;
668 end
669
670 % Plot the MES (with the option to plot the true log-normal density)
671 function [xx,yy]=plotMES(data,lamStar,eta, sigma)
672 k = data(1:end-1,1);
673 K = data(end,1);
674 m=length(k);
675 xx=linspace(min(k),max(k),(7+1)*(m-1)+1);
676 xx=[xx,linspace(max(k),K,10)];
677 yy=zeros(1,length(xx));
678 h_indAll=1:(m+1);
679 if  any(eta==0)
680     temp=lamStar*h_function(xx,data,h_indAll,eta);
681     index=find(abs(temp)>1e-5);
682     yy(index)=exp(temp(index)-1);
683 else
684     yy=exp(lamStar*h_function(xx,data,h_indAll,eta)-1);
685 end
686 b=MEScdf(K,data,lamStar,eta);
687 yy = yy/b;
688 plot(k,zeros(size(k)),'ko');
689 hold on
690 plot(xx,yy,'b')
691 plot([0,min(k),xx(1)],[0 0 yy(1)],'b:');
692 xlabel('stock price x')
693 ylabel('p(x)')
694 if nargin == 4
695     [~,~,~,K,mu,sd,~]=getParam(sigma);
```

```
    fx=lognpdf(xx,mu,sd)/(logncdf(K,mu,sd)-logncdf(min(k),mu,sd));
    plot(xx,fx,'k--')
    plot([0,min(k),xx(1)],[0 0 fx(1)], 'k:');

    legend('MES','true pdf','strike prices')
end
hold off
end

% Estimate the integrated squared error from int_sqr_err function
function out=getInt_sqr_err(data,lamStar,sigma,eta)
[~,~,~,~,mu,sd,~]=getParam(sigma);
K=data(end,1);
k=data(:,1);
xx=linspace(min(k),K,100000);
deltaX=xx(2)-xx(1);

[y1,y2]=int_sqr_err(xx,data,lamStar,mu,sd,eta);
out=sum((y1-y2).^2*deltaX);
end

% Obtains the integrand of the integrated squared error
function [y1,y2]= int_sqr_err(x,data,lamStar,mu,sd,eta)
K=data(end,1);
k = data(1:end-1,1);
h_indAll=1:size(data,1);
y1 = zeros(1,length(x));
for i = 1: length(x)
    y1(i)=exp(lamStar*h_function(x(i),data,h_indAll,eta)-1);
end
y1 = y1/MEScdf(K,data,lamStar,eta);
y2 = lognpdf(x,mu,sd)/(logncdf(K,mu,sd)-logncdf(min(k),mu,sd));
end

% Plot the ISE, MISE, and proportion of convergence graphs
function [mise,se,pise]=isePlots(ise,N, alphaMat, pMat)
if nargin == 3
    pMat = 1e-3;
end
ise(ise==0)=NaN;
if length(alphaMat)==1 || length(pMat) == 1
    if length(alphaMat) > length(pMat)
        parMat = alphaMat;
        parName = sprintf('regularisation parameter, a');
    elseif length(pMat) > length(alphaMat)
        parMat = pMat;
        parName = sprintf('mixing parameter, p');
    else
        fprintf('Make sure one of the alphaMat or pMat needs to be a scalar');
    end
end
pise = zeros(1,size(parMat,2));
mise = pise;
se = pise;
```

```
for i = 1:length(parMat)
    temp=ise(:,i);
    temp(isnan(temp))=[];
    n = length(temp);
    pise(i)=n/N;
    mise(i) = mean(temp);
    if n <= 1
        se(i)=NaN;
    else
        se(i)=1.96*std(temp)/sqrt(n);
    end
end
figure;
subplot(2,2,[1 2])
boxplot(ise,'labels',parMat);
xlabel(parName);
ylabel('ISE');
title('Boxplots of ISE')

subplot(2,2,3)
myerrorbar(mise, se, parMat)
xlabel(sprintf('index of the %s',parName));
ylabel('MISE');
title('MISE and 95% confidence interval')

subplot(2,2,4);
plot(parMat,pise);
hold on
plot(parMat,pise,'ro');
xlabel(parName);
ylabel('Proportion of Convergence');
title('Proportion of Convergence')
end

% Constraints for the polyhedral set projection method
function [c,ceq] = nonlcon(x,data)
B=getB(data);
m=size(B,1)-1;
u1=ones(m+1,1);
epsilon=0;
c = -B*[1;x] + epsilon*u1;
ceq = u1'*B*[1;x] -1;
end
```

Bibliography

[1] M. Avellaneda. The Minimum-Entropy Algorithm and Related Methods for Calibrating Asset-Pricing models. *Doc. Math.*, 3:545—-563, 1998.

[2] M. Baxter and A. Rennie. *Financial Calculus*. Cambridge Univ Press, 1997.

[3] U. Benzion, S. Anan, and J. Yagil. Box Spread Strategies and Arbitrage Opportunities. *The Journal of Derivatives*, pages 47–62, 2005.

[4] T. Bjork. *Arbitrage Theory in Continuous Time*. Oxford University Press, 1998.

[5] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy*, 81(3):637–654, 1973.

[6] O. Bondarenko. Recovering Risk-Neutral Densities: A New Nonparametric Approach. *Working Paper, University of Illinois, California*, 2000.

[7] J. Borwein, R. Choksi, and P. Maréchal. Probability Distributions of Assets Inferred from Option Prices via the Principle of Maximum Entropy. *Society for Industrial and Applied Mathematics*, 14(2):464–478, 2003.

[8] C. Bose and R. Murray. Maximum Entropy Estimates for Risk-neutral Probability Measures with Non-Strictly-Convex Data. *Working Paper*, pages 1–26, 2011.

[9] D. Breeden and R. Litzenberger. Prices of State-Contingent Claims Implicit in Option Prices. *Journal of Business*, 51(4):621–651, 1978.

[10] P. Buchen and M. Kelly. The Maximum Entropy Distribution of An Asset Inferred from Option Prices. *Journal of Financial and Quantitative Analysis*, 31(01):143–159, 1996.

[11] J. Cox and A. Ross. The Valuation of Options for Alternative Stochastic Processes. *Journal of Financial Economics*, 3(20292):145–166, 1976.

[12] D. Ehrman. *The Handbook of Pairs Trading: Strategies Using Equities, Options, and Futures*. John Wiley & Sons, 2006.

[13] R. Eubank. *Nonparametric Regression and Spline Smoothing*. Marcel Dekker, 2nd edition, 1999.

[14] S. Figlewski. What Does an Option Pricing Model Tell Us About Option Prices? *Financial Analysts Journal*, 45(5):12–15, 1989.

[15] W. Guo. Maximum Entropy in Option Pricing: A Convex-Spline Smoothing Method. *Journal of Futures Markets*, 21(9):819–832, 2001.

[16] P. Hagan, A. Lesniewski, and D. Woodward. Probability Distribution in the SABR Model of Stochastic Volatility. *Working Paper*, 2005.

[17] S. Herzel. Arbitrage Opportunities on Derivatives: A Linear Programming Approach. *Dynamics of Continuous, Discrete, and Impulsive Systems*, 12(4):589–606, 2005.

[18] S. Heston. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *Review of Financial Studies*, 1993.

[19] J. Hull. *Options, Futures and Other Derivatives*. Pearson Prentice Hall, 7th edition, 2009.

[20] J. Jackwerth and M. Rubinstein. Recovering Probability Distributions from Option Prices. *Journal of Finance*, pages 1611–1631, 1996.

[21] E. Jaynes. Information Theory and Statistical Mechanics. *Statistical Physics. Brandeis Lectures*, 3:160–185, 1963.

[22] F. Longstaff. Option Pricing and the Martingale Restriction. *Review of Financial Studies*, 8(4):1091–1124, October 1995.

[23] R. Merton. Theory of Rational Option Pricing. *The Bell Journal of Economics and Management*, 4(1):141–183, 1973.

[24] A. Monteiro, R. Tütüncü, and L. Vicente. Recovering Risk-Neutral Probability Density Functions from Options Prices Using Cubic Splines and Ensuring Nonnegativity. *European Journal of Operational Research*, 187(2):525–542, June 2008.

[25] MC Newman. Regression Analysis of Log-Transformed data: Statistical Bias and its Correction. *Environmental Toxicology and Chemistry*, 1993.

[26] J. Orozco-Rodriguez and F. Santosa. Estimation of Asset Distributions from Option Prices: Analysis and Regularization. *SIAM Journal on Financial Mathematics*, 3:374–401, 2012.

[27] R. Rockafellar. Integrals which are Convex Functionals. *Pacific J. Math*, 24(3):525–540, 1968.

[28] S. Ross. Options and Efficiency. *The Quarterly Journal of Economics*, 90(1):75–89, 1976.

[29] M. Rubinstein. Implied Binomial Trees. *Journal of Finance*, 49(3):771–818, 1994.

[30] C. Shannon. The Mathematical Theory of Communication. *M.D. Computing : Computers in Medical Practice*, 14(4):306–17, 1948.